EOARD project #033060
Speaker verification using a dynamic, 'articulatory' segmental hidden Markov model

Final Progress Report, November 2004

Ying Liu and Martin Russell

Department of Electronic, Electrical and Computer Engineering
The University of Birmingham, Edgbaston, Birmingham B15 2TT, UK

`m.j.russell@bham.ac.uk; liuy2@eee-fs7.bham.ac.uk`

November 2004

Abstract

This is the final report for EOARD project #033060 "Speaker verification using a dynamic, 'articulatory' segmental hidden Markov model".

A segmental HMM is a HMM whose states are associated with sequences of acoustic feature vectors rather than individual vectors. This report describes the results of experiments in which such a model is applied to text-dependent and -independent speaker-detection on the YOHO and Switchboard corpora, respectively. Text-dependent speaker verification results on YOHO using a simple segmental HMM show a 44% reduction in false acceptances compared with a conventional HMM. A type of 'segmental GMM' is then described for text-independent speaker detection. In order to apply this model to the NIST 2003 single-speaker test set, various techniques are developed to reduce its computational load. A range of experiments are then reported which investigate the utility of different aspects of this model for text-independent speaker-detection. From these experiments we have been unable to demonstrate a benefit, in terms of text-independent speaker-detection accuracy, from the use of dynamic segment models corresponding to linear trajectories with non-zero slope. Consequently we have also been unable to demonstrate any benefit from the use of longer segments. Thus there is little evidence from these experiments that non-stationary sections of a speech signal contain important individual differences which can be exploited for speaker-detection. If this is true, it goes some way towards explaining the success of GMM-based approaches. We conclude that further work, to determine definitively the contribution of non-stationary segments to speaker-detection, is needed.

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. REPORT DATE ***(DD-MM-YYYY)*** 23-12-2004	**2. REPORT TYPE** Final Report	**3. DATES COVERED** *(From – To)* 1 October 2003 - 17-Jun-05

4. TITLE AND SUBTITLE Speaker Verification Using A Dynamic, 'Articulatory' Segmental Hidden Markov Model	**5a. CONTRACT NUMBER** FA8655-03-1-3060
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Dr. Martin J Russell	**5d. PROJECT NUMBER**
	5d. TASK NUMBER
	5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Birmingham The University of Birmingham Birmingham B15 2TT United Kingdom	**8. PERFORMING ORGANIZATION REPORT NUMBER** N/A
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) EOARD PSC 802 BOX 14 FPO 09499-0014	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) SPC 03-3060

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report results from a contract tasking University of Birmingham as follows: The Grantee will investigate whether an improved model of speech dynamics and the use of an 'articulatory' representation improves speaker recognition performance. The grantee has developed a new type of 'Multiple-level' segmental Hidden Markov Model (MSHMM) for automatic speech recognition, in which the relationship between the symbolic and acoustic representations of a speech signal is regulated by an intermediate, 'articulatory' representation. States of the model define trajectories in the articulatory domain, which are transformed into the acoustic domain using an 'articulatory-to-acoustic' mapping. Comparison is then made with an unknown acoustic speech pattern. A non-linear 'acoustic-to-articulatory' mapping model has many advantages for speech recognition. Speech dynamics can be modelled directly in the articulatory domain, production strategies used in different speaking styles can be characterised, and physical differences between speakers can be represented explicitly. In principle these advantages also apply to speaker recognition. MSHMMs will enable the importance of speech dynamics and the utility of articulatory parameters for speaker recognition to be studied. Hence the goal of this proposal is to apply MSHMMs to speaker verification.

15. SUBJECT TERMS

EOARD, Speech Processing, Operator-Machine Interface, Information Assurance

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18, NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT UNCLAS	**b. ABSTRACT** UNCLAS	**c. THIS PAGE** UNCLAS	UL	68	VALERIE E. MARTINDALE, Lt Col, USAF
					19b. TELEPHONE NUMBER *(Include area code)* +44 (0)20 7514 4437

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 23-12-2004	**2. REPORT TYPE** Final Report	**3. DATES COVERED** *(From – To)* 1 October 2003 - 17-Jun-05

4. TITLE AND SUBTITLE

Speaker Verification Using A Dynamic, 'Articulatory' Segmental Hidden Markov Model

5a. CONTRACT NUMBER
FA8655-03-1-3060

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Dr. Martin J Russell

5d. PROJECT NUMBER

5d. TASK NUMBER

5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Birmingham
The University of Birmingham
Birmingham B15 2TT
United Kingdom

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

EOARD
PSC 802 BOX 14
FPO 09499-0014

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
SPC 03-3060

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report results from a contract tasking University of Birmingham as follows: The Grantee will investigate whether an improved model of speech dynamics and the use of an ‘articulatory’ representation improves speaker recognition performance. The grantee has developed a new type of ‘Multiple-level’ segmental Hidden Markov Model (MSHMM) for automatic speech recognition, in which the relationship between the symbolic and acoustic representations of a speech signal is regulated by an intermediate, ‘articulatory’ representation. States of the model define trajectories in the articulatory domain, which are transformed into the acoustic domain using an ‘articulatory-to-acoustic’ mapping. Comparison is then made with an unknown acoustic speech pattern. A non-linear ‘acoustic-to-articulatory’ mapping model has many advantages for speech recognition. Speech dynamics can be modelled directly in the articulatory domain, production strategies used in different speaking styles can be characterised, and physical differences between speakers can be represented explicitly. In principle these advantages also apply to speaker recognition. MSHMMs will enable the importance of speech dynamics and the utility of articulatory parameters for speaker recognition to be studied. Hence the goal of this proposal is to apply MSHMMs to speaker verification.

15. SUBJECT TERMS
EOARD, Speech Processing, Operator-Machine Interface, Information Assurance

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18, NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT UNCLAS	**b. ABSTRACT** UNCLAS	**c. THIS PAGE** UNCLAS	UL	68	VALERIE E. MARTINDALE, Lt Col, USAF **19b. TELEPHONE NUMBER** *(Include area code)* +44 (0)20 7514 4437

Contents

1 Introduction

This is the final report for EOARD project #033060 "Speaker verification using a dynamic, 'articulatory' segmental hidden Markov model", which started on 1$^{\text{st}}$ *October 2003. The report describes technical progress which has been made, discusses the speaker verification results, and outlines possible future work.*

Some recent work in speech recognition conducted as part of the 'Balthasar' project[1] (Russell and Jackson 2005) at the University of Birmingham has resulted in a class of novel, multiple-level Segmental Hidden Markov Models (MSHMM) in which the relationship between symbolic and acoustic representations of a speech signal is regulated by an intermediate 'articulatory' layer (figure 1). Each state of the model is associated with variable-duration trajectories in the 'articulatory' space, which are mapped into the acoustic space using one or more 'articulatory-to-acoustic' mappings. Comparison with unknown speech data, for the purposes of probability calculations, takes place in the acoustic space. A similar approach has been studied by Deng and Ma (Deng and Ma 2000).

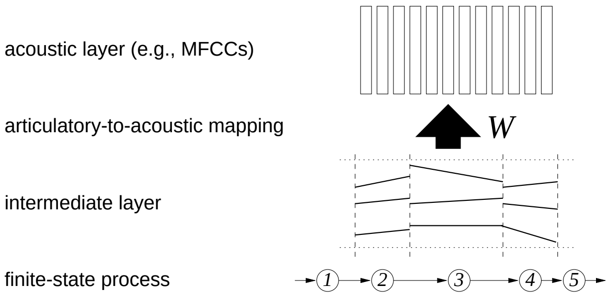

acoustic layer (e.g., MFCCs)
articulatory-to-acoustic mapping
W
intermediate layer
finite-state process
1
2
3
4
5

Figure 1: A segmental models that uses linear trajectories in an intermediate space.

Such an approach has many potential advantages for speech pattern processing. For example, in acoustic representations of speech (derived from short-term log-power spectra) articulator dynamics are manifested indirectly, often as movement between, rather than within, frequency bands. Intuitively, therefore, it would be much better to model dynamics directly, in an articulatory-based representation. Also, by incorporating an articulatory representation (or at least one which is more closely related to an articulatory representation than conventional spectrum-based acoustic representations), it may be possible to characterise the production strategies that give rise to variability in fluent, conversational speech. Thus it was hoped that such a model would improve speech recognizer performance by modelling the underlying mechanisms that cause variability, rather than relying solely on generic statistical modelling techniques.

[1] The "Balthasar" project was funded by EPSRC grant GR/M87146 "An integrated multiple-level statistical model for speech pattern processing" (see http://web.bham.ac.uk/p.jackson/balthasar)

This approach should also have benefits for speaker detection, and the goal of this project is to apply M-SHMMs to that problem. The benefits should fall into two categories:

- Those which derive from the incorporation of an explicit 'articulatory-related' representation into a statistical model, and
- Those which derive from the improved modelling of speech dynamics and duration which results from the use of a segmental framework.

In the first category, inter-speaker differences which result from physiological factors, such as the differences between an adult's vocal tract and that of a child, should be represented explicitly in the articulatory layer rather than indirectly through their acoustic correlates. The model should also enable individual differences in the articulatory strategies used by a speaker during speech production to be exposed and modelled explicitly. Furthermore, provided that the articulatory-based representation is sufficiently compact, there should also be significant advantages for speaker adaptation from limited amounts of data, since less data will be needed to train the smaller number of parameters. As an illustration, in (Russell and Jackson 2005) it is shown that a triphone M-SHMM system with an intermediate representation based on just 3 formant frequencies can achieve better phone classification results on TIMIT, while at the same time having 25% fewer parameters than the conventional system. Of course, in order to realise this benefit fully it will be necessary to extend speaker adaptation techniques, such as MAP (Gauvain and Lee 1994) or MLLR (Leggetter and Woodland 1995) to the articulatory layer of a M-SHMM. This is currently being studied in a separate PhD project.

The second type of potential benefit derives from improved model of speech dynamics and duration. The model should be able to capture individual differences in non-stationary speech segments which might otherwise be swamped by large variance due to the HMM piecewise stationarity assumption. Thus it is plausible that such a model will improve our understanding of inter-speaker differences, and hence improve speaker detection performance, by modelling some of the underlying mechanisms that give rise to intra- and inter-speaker differences. It would also be possible to determine whether non-stationary speech segments are any more or less useful for speaker detection than stationary segments.

The speaker-detection experiments described in this report focus on the second set of factors. In other words we apply simple linear-trajectory segmental HMMs in which the intermediate representation is absent, to speaker detection. These M-SHMMs are equivalent to the 'Fixed Trajectory' segmental HMMs described in (Holmes and Russell 1999). In practice, this type of segmental HMM is realised in the 'SEGVit' software toolkit by setting the intermediate space equal to the acoustic space, and by setting the 'articulatory-to-acoustic' mapping to be the identity mapping I.

The report is organised as follows. In section 2 we present some of the relevant background results on the application of M-SHMMs to phone recognition from the earlier 'Balthasar' project. Section 3 is a brief description of relevant aspects of the theory of M-SHMMs. The main part of the report presents results of two sets of experiments, namely text-dependent speaker detection experiments on the YOHO corpus (section 4), and text-independent speaker-detection experiments on a subset of the 2003 NIST speaker recognition evaluation test set (section 5). The first set of experiments were also reported at the 2004 'Odyssey' Speaker Recognition Workshop in Toledo, Spain, in June 2004 (Liu, Russell, and Carey 2004). Our conclusions, and suggestions for further work, are set out in section 8.

2 Relevant results from the 'Balthasar' Project

In this section we review some of the relevant results on general M-SHMMs from the phone classification experiments reported in (Russell and Jackson 2005).

A potential problem with the type of multiple-level model described in the previous section is that any advantages which are gained by the introduction of an intermediate layer may be compromised by inadequacies of the articulatory representation or articulatory-to-acoustic mapping, or theoretical compromises made for mathematical or computational tractability.

In (Russell and Jackson 2005) M-SHMMs were studied in which the intermediate representation is based on the control parameter set for the Holmes-Mattingley-Shearme (HMS) parallel formant synthesiser (Holmes, Mattingly, and Shearme 1964). Three different formant-based intermediate parameterisations were considered:

- 3FF - the first three formant frequencies, *F1, F2* and *F3*. It is clear that these three parameters alone do not contain sufficient information to reconstruct a short term spectrum (or MFCC vector) unambiguously. For example, there is no data concerning formant amplitudes

- 3FF+5BE - the first three formant frequencies plus 5 band energies

- 12PFS - the complete set of 12 Holmes-Mattingley-Shearme parallel formant synthesiser control parameters (Holmes, Mattingly, and Shearme 1964). Experiments conducted by Holmes in the early 1970's demonstrated that these parameters, if chosen correctly, are sufficient to synthesise natural sounding speech (Holmes 1973).

In all of these experiments speech dynamics were modelled using linear trajectories, and the articulatory-to-acoustic mapping was realised as a set of one or more linear mappings (Russell and Jackson 2005).

It is easy to see that a linear 'articulatory-to-acoustic' mapping is not sufficient for speech pattern modelling (Richards and Bridle 1999). For example, consider the case where speech is represented in the acoustic domain as the output of a set of D_2 uniformly-spaced band-pass filters spanning frequencies up to 4 kHz, and f is a hypothetical ‘formant’ trajectory, with unit amplitude, whose frequency increases linearly from 100 Hz to 4 kHz. The corresponding trajectory in acoustic space is a complex path over the surface of the D_2 dimensional unit sphere, which passes through each of the axes in turn. Such a trajectory clearly cannot be realised as the image of f under a single linear mapping. However, previous experience has shown that even relatively small deviations from the conventional HMM framework can result in significant difficulties and poor performance. Therefore it was judged that a proper understanding of the issues which arise in the implementation of a system with linear transformations is essential before attempting to deal with more complex non-linear systems.

The key results reported in (Russell and Jackson 2005) are:

- There is a theoretical upper bound on the performance of a linear M-SHMM, which is better than that obtained with a comparable conventional HMM
- This upper bound can be attained by appropriate choice of ‘articulatory’ representation and articulatory-to-acoustic mappings
- There is a trade-off between the dimension of the ‘articulatory-based’ space and the number of different mappings which make up the piecewise-linear ‘articulatory-to-acoustic’ mapping. For example, optimal performance can be achieved by using all 12 HMS synthesiser control parameters and a single (phone-independent) linear mapping, or by using fewer parameters but more, phone-dependent, mappings (Russell and Jackson 2005)

The significance of this result, in general, is that it provides a solid theoretical foundation for the development of richer classes of multi-level models, which include non-linear models of dynamics, alternative articulatory representations, sets of non-linear articulatory-to-acoustic mappings, and integrated optimisation schemes that support unsupervised learning of the trajectory, intermediate representation and mapping parameters. Moreover, these speech recognition results also motivate the application of M-SHMMs to speaker detection.

3 Overview of the theory of M-SHMMs

The purpose of this section is to explain the basic theory of multiple-level, trajectory-based segmental HMMs (M-SHMMs) and the simpler fixed trajectory segmental HMMs used in our experiments. Full details are presented in (Russell and Jackson 2005) and (Holmes and Russell 1999). This section should be skipped by anyone who is familiar with these papers.

3.1 Definitions

As explained earlier, a M-SHMM is a particular type of *segmental* hidden Markov model (SHMM) (Ostendorf, Digalakis, and Kimball 1996). In other words, the states of a MSHMM are associated with sequences of feature vectors, or *segments*, rather than individual vectors. The model is called 'multiple-level' because it considers two levels of representation of a speech signal: a D_1 dimensional 'articulatory' space $\mathcal{I}$ and a D_2 dimensional acoustic space $\mathcal{A}$. In (Russell and Jackson 2005) the 'articulatory' and acoustic spaces are based on formants and Mel-Frequency Cepstral Coefficients (MFCCs), respectively.

3.2 The multiple-level, linear-trajectory segment model

A state σ_i of a M-SHMM is identified with a variable duration linear trajectory in $\mathcal{I}$ which is mapped into $\mathcal{A}$ by a linear 'articulatory-to-acoustic' mapping. A state is parameterised by two D_1 dimensional (articulatory) vectors, namely the mid-point vector $\mathbf{c}_i$ and slope vector $\mathbf{m}_i$, a $D_2 \times D_2$ (acoustic) covariance matrix V_i, and a linear 'articulatory-to-acoustic' mapping $W_i : \mathcal{I} \rightarrow \mathcal{A}$. A trajectory $\mathbf{f}$ of length τ is defined by:

$$\mathbf{f}_i(t) = (t - \bar{t})\mathbf{m}_i + \mathbf{c}_i \tag{1}$$

where $\bar{t} = (\tau+1)/2$, and the function of W_i is to map this 'articulatory' trajectory in $\mathcal{I}$ into the acoustic space $\mathcal{A}$. If $Y_1^\tau = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_\tau]$ is a sequence of acoustic feature vectors in $\mathcal{A}$, then the probability (density) of Y_1^τ given state σ_i is given by:

$$p(Y_1^\tau|\sigma_i) = b_i(Y_1^\tau) = d_i(\tau) \prod_{t=1}^{\tau} \mathcal{N}\left(\mathbf{y}_t; W_i(\mathbf{f}_i(t)), V\right), \tag{2}$$

where $d_i(\tau)$ is the probability that state σ_i emits a segment of length τ, and $\mathcal{N}\left(\mathbf{y}_t; W_i(\mathbf{f}_i(t)), V_i\right)$ is a D_2 dimensional Gaussian probability density function (PDF) with mean $W_i(\mathbf{f}_i(t))$ and covariance matrix V_i (it is assumed tha V_i is diagonal).

In the special case where $\mathcal{I} = \mathcal{A}$ and W_i is the identity matrix, this reduces to a Fixed Trajectory Segmental HMM (Holmes and Russell 1999) and equation 2 becomes:

$$p(Y_1^\tau|\sigma_i) = b_i(Y_1^\tau) = d_i(\tau) \prod_{t=1}^{\tau} \mathcal{N}\left(\mathbf{y}_t; \mathbf{f}_i(t), V\right), \tag{3}$$

3.3 The segmental Viterbi decoder

Let $\mathcal{M}$ be an S-state MSHMM (for simplicity, it is assumed that the probability of a transition from σ_i to state σ_j is zero unless $j \geq i$). Suppose that the acoustic sequence $Y_1^T = [\mathbf{y}_1, \ldots, \mathbf{y}_T]$ corresponds to several states/segments. Then $\mathcal{M}$ can only explain Y via a state sequence $\mathbf{x} = [x_1, \ldots, x_T]$, which can be written in the form $\mathbf{x} = [d_1 \otimes z(1), \ldots, d_L \otimes z(L)]$, For each $l \in \{1, \ldots, L\}$, $z(l) = \sigma_i$ for some $i \in \{1, \ldots, S\}$, and $d_l \otimes z(l)$ represents a duration d_l spent in state $z(l)$. Thus, the joint density has the form,

$$p\left(Y, \mathbf{x}|\mathcal{M}\right) = \pi_{z(1)} b_{z(1)}\left(Y_{t_1}^{t_2-1}\right) \prod_{l=2}^{L} a_{z(l-1),z(l)}\, b_{z(l)}\left(Y_{t_l}^{t_{(l+1)}-1}\right), \tag{4}$$

where $\pi_{z(1)}$ is the probability that the state sequence begins in state $z(l)$; $b_{z(l)}$ denotes the acoustic segment pdf associated with state $z(l)$; $a_{z(l-1),z(l)}$ denotes the transition probability from $z(l-1)$ to $z(l)$; t_l is the time at which the state sequence $\mathbf{x}$ enters state $z(l)$, and $t_{L+1} = T + 1$.[2]

A simple extension of the segmental Viterbi decoder (see, for example, Holmes and Russell 1999) can be used to compute the optimal state sequence $\hat{\mathbf{x}}$ for a given sequence of acoustic vectors Y_1^T and model $\mathcal{M}$, such that

$$\hat{p}\left(Y|\mathcal{M}\right) = p\left(Y, \hat{\mathbf{x}}|\mathcal{M}\right) = \max_{\mathbf{x}} p\left(Y, \mathbf{x}|\mathcal{M}\right). \tag{5}$$

For completeness, a brief description of the segmental Viterbi decoder is included. By analogy with the notation for the forward probability used in the case of a conventional HMM (see, for example, Holmes and Holmes 2001), let

$$\hat{\alpha}_j(t) = max_{x_1,\ldots,x_{t-1}} p(\mathbf{y}_1, \ldots, \mathbf{y}_t; x_t = s_j, x_{t+1} \neq s_j). \tag{6}$$

The final condition, $\mathbf{x}_{t+1} \neq s_j$ is included to ensure that only segments which are complete at time t are considered. Then, it can be shown that,

$$\hat{\alpha}_j(t) = \max_i \max_\tau \begin{cases} \pi_i\, b_i(Y_1^\tau) & \text{for } t = \tau \\ \hat{\alpha}_i(t-\tau)\, a_{i,j}\, b_j(Y_{t-\tau+1}^t) & \text{for } t > \tau \end{cases} \tag{7}$$

where $1 \leq \tau \leq \tau_{\max}$ and π_i is the probability that the state sequence begins in the i^{th} state. The requirement in Equation 7 to optimise over all possible segment durations, τ, and to evaluate segmental state output probabilities, $b_j(Y_{t-\tau+1}^t)$, leads to a substantial increase in computational load relative to the normal Viterbi decoder. As in conventional Viterbi decoding for continuous speech recognition, this algorithm is applied to a single, integrated MSHMM in which the individual word- or phone-level MSHMMs are connected according to a grammar (Bridle et al. 1983). Thus, in the case of phone recognition and a bigram language model, the result of decoding is the sequence of phones $[\rho_1, \ldots, \rho_\Phi]$ and phone boundaries $[t_1, \ldots, t_\Phi]$ such that the joint probability,

$$\hat{p}\left(Y; t_1, \ldots, t_\Phi; \rho_1, \ldots, \rho_\Phi\right) = \hat{p}\left(Y_1^{t_1}|\rho_1\right) \prod_{\phi=2}^{\Phi} \left(P\left(\rho_\phi|\rho_{\phi-1}\right)^{\lambda_1} \hat{p}\left(Y_{t_{\phi-1}+1}^{t_\phi}|\rho_\phi\right) \lambda_2\right), \tag{8}$$

[2] The introduction of symbol $z(l)$ to denote a state simplifies subsequent notation, in particular Eq. 4. In the symbol x_t, the time index t is in synchrony with the observation sequence $\mathbf{y}_t$; whereas for $z(l)$, the index l is in synchrony with the state transitions. Unlike in a conventional HMM, the two are not generally the same in a SHMM.

is maximised. Here, λ_1 is the Language Model Scale Factor (LMSF) and λ_2 is the Token Insertion Penalty (TIP).

Our current software uses a single implementation of the segmental Viterbi decoder for embedded and non-embedded training, phone classification and phone recognition. This is achieved by introducing a time-indexed array of breakpoints that specify, at each time t, whether a phone boundary is obligatory, possible or illegal. An additional parameter, τ_{max}, specifies the maximum permissible segment length.

4 Text-Dependent Speaker Verification

The most straightforward application of M-SHMMs to speaker recognition is text-dependent speaker verification (TD-SV). This is because a conventional TD-SV system typically uses phone-level or word-level HMMs, which can simply be replaced by the corresponding M-SHMMs.

Suppose that a sequence of acoustic feature vectors $Y = [y_1, ..., y_N]$ is claimed to result from subject S speaking a text Q. The decision whether to accept or reject this claim is based on the likelihood ration:

$$L(S) = \frac{p(Y|S, Q)}{p(Y|Q)} \tag{9}$$

where $p(Y|S, Q)$ is computed using a set of word- or phone-level models for speaker S, configured to represent the text Q, and $p(Y|Q)$ is calculated using a set of speaker-independent models configured to represent Q.

Our experiment used the YOHO (Higgins 1990) and TIMIT (Garofolo et al. 1993) speech corpora. As this was an initial exploration of the application of M-SHMMs to speaker recognition, we considered a Fixed Trajectory Segmental HMM, in which there is no intermediate 'articulatory-based' representation (i.e. $\mathcal{I} = \mathcal{A}$ and $W_i = I$). Thus the experiment focusses on the utility of improved modelling of duration and dynamics for speaker recognition (and not on the utility of introducing an intermediate, 'articulatory-based' representation).

4.1 Experimental Method

4.1.1 The TIMIT and YOHO speech corpora

The YOHO corpus comprises recordings of 138 subjects speaking connected digit-sequence phrases in an office environment. It was chosen because of its established use in text-dependent speaker verification (Higgins 1990). The speech in the YOHO corpus is sampled at 8kHz.

The TIMIT corpus is very well-known, and comprises recordings of read speech. TIMIT is labeled at the phone level, and is therefore particularly useful for building phone-level acoustic models. Speech in the training component of the TIMIT corpus was downsampled to 8kHz sampling rate, for compatibility with YOHO.

4.1.2 Acoustic parameterisation

All of the data was parameterised using the Hidden Markov Model Tool Kit (HTK) tool 'HCopy' (Young, Odell, Ollason, Valtchev, and Woodland 1997). Each file is represented as a sequence of 13 dimensional feature vectors, one every 10ms, comprising MFCCs 1 to 12 plus energy.

No Δ or Δ^2 parameters were used in any of the experiments. In fact, we have not yet used Δ or Δ^2 parameters in any of our previous M-SHMM based speech recognition experiments. This is because part of the motivation for the development of MSHMMs is to obtain a better model of speech dynamics and thereby obviate the need for these parameters.

In a conventional HMM, the assumptions that the underlying structure of a speech segment is stationary, and that the static, Δ and Δ^2 parameters are non-zero, are clearly inconsistent. A trajectory-based model could overcome this inconsistency: for such a model to incorporate non-zero Δ parameters, linear trajectories would be needed, while one which included non-zero Δ and Δ^2 would need quadratic trajectories. The issues raised by including dynamic features in a conventional HMM are discussed in (Bridle 2004).

4.1.3 Construction of initial acoustic models using TIMIT

The TIMIT training data set was used to estimate the initial parameters for matching sets of context-sensitive triphone HMMs and MSHMMs. The HMMs and the M-SHMMs were built using the SEGVit M-SHMM software toolkit developed at the University of Birmingham. In both cases, monophone models with three emitting states were constructed first, and then used to seed a set of triphone models. The triphone model set was defined using a simple 'backoff' strategy whereby a triphone model was constructed if and only if 30 or more examples of that triphone context occurred in the training data, otherwise the triphone was replaced by a biphone (if 30 or more examples of the biphone context occurred in the training data) or a monophone. This is the 1400 triphone model set from (Russell and Jackson 2005). In the case of MSHMMs, the maximum segment duration τ_{max} was set to 15 and the duration probability mass functions d_i were non-parametric (Ferguson duration model (Ferguson 1980)).

46 of these triphones were needed to model the 102 cross-word triphones in the YOHO corpus.

The states of the conventional HMMs are associated with single Gaussian densities. This is for compatibility with the M-SHHM system, which currently cannot accommodate multiple-component Gaussian mixture densities. The conventional monophone HMMs were intialised and reestimated using the HTK tools 'HInit' and 'HRest' respectively (Young, Odell, Ollason, Valtchev, and Woodland 1997). These conventional monophone HMMs were also used to seed the monophone M-SHMMs, by setting the M-SHMM state mean and variance vectors equal to the corresponding HMM state mean vectors, and setting the

M-SHMM state slope vectors equal to zero. The 'self-loop' state-transition probabilities were set to zero in the case of the M-SHMMs, but were non-zero for the conventional HMMs.

4.1.4 Construction of background and speaker-dependent models using YOHO

Models for those triphones which occur in the YOHO data were used to seed speaker-independent sets of YOHO HMMs and M-SHMMs, which were trained on all of the data from 20 of the subjects (10 female, 10 male) in the YOHO corpus. These models formed the HMM and M-SHMM Background Models (BMs). The HMM and M-SHMM BMs were each trained using 20 iterations of Baum-Welch (HTK) and Viterbi-based (SEGVit) training respectively.

The remaining 118 subjects were used as test subjects. For each of these subjects, 96 files were used to train speaker-dependent HMMs and MSHMMs. As with the BMs, the HMM and M-SHMM SDMs were trained using 20 iterations of Baum-Welch and Viterbi-based training, respectively. The remaining 20 files were split into 5 test sets, each containing 4 speech files. A single experiment consisted of comparing 1 such test set with a speaker dependent model and BM. Thus, for each system, the number of 'authorised user' trials is $118 \times 5 = 590$, and the number of 'impostor' experiments is $118 \times 117 \times 5 = 69030$.

4.2 Results of text-dependent speaker detection experiments on YOHO

The results of the text-dependent speaker verification experiments are shown as DET curves in Appendix A (figure 4). The lower-bound of 0.17% in the figure for the false rejection probablity equates to a single rejection out of the 590 'authorised user' trials. It is likely that this results from incorrectly labelled data. Because of this small number of errors there is no opportunity to compare the HMM and M-SHMM systems in terms of false rejection rates on this data set. Both systems achieve an optimal false rejection rate of 0.5%.

The false acceptance rates for the HMM and MSHMM systems provide a more useful comparison. At the optimal points these are 0.52% for the HMM system and 0.29% for the M-SHMM system, corresponding to 359 and 200 false acceptances, respectively. This equates to a 44% reduction in the number of false acceptances by using the M-SHMM system, relative to the conventional HMM-based system.

4.3 Summary of Text-Dependent Verification Results

In summary, there is some evidence from this experiment that a M-SHMM-based text-dependent speaker verification system can outperform a conventional HMM-based system. This is illustrated by the reduction in false acceptance errors. However, particularly in the case of false rejection errors, the resolution of this test is not sufficiently fine to draw clear conclusions. Therefore it was decided that a more difficult speaker-detection task should be attempted, namely text-independent speaker detection on the Switchboard corpus.

5 Text-Independent Speaker Verification

5.1 A 'segmental GMM'

Although many different approaches to text-independent speaker detection have been tried, the most successful approach to-date is undoubtedly probabilistic classification using Gaussian Mixture Models (GMMs) (Reynolds 1992). As in the text-dependent case, to test the hypothesis that a sequence of acoustic feature vectors $Y = [y_1, y_2, ..., y_N]$ was spoken by a talker S, the likelihood ratio

$$L(S) = \frac{p(Y|S)}{p(Y)} \tag{10}$$

is computed and compared with a pre-determined threshold T. The probability $p(Y)$ is computed using a 'Background Model' (BM) or 'General Speaker Model' (GSM), which is a GMM trained on acoustic feature vectors corresponding to speech produced by a large population of talkers. The value of $p(Y|S)$ is computed using a 'speaker model' for speaker S, which is a GMM trained on acoustic feature vectors derived from speech produced by S (or, more normally, adapted from the BM). The quantity $L(S)$ in equation (10) is an approximation to the posterior probability of S given the data Y, where the prior probability $P(S)$ of speaker S is ignored. The score $L(S)$ is often normalised to allow the same threshold to be used for all talkers (Auckenthaler, Carey, and Lloyd-Thomas 2000).

In order to compare conventional methods with a M-SHMM-based method for text-independent speaker detection, it is therefore natural to attempt to construct a segmental HMM version of a conventional GMM based speaker recognition system.

In a GMM-based system:

- A speech signal is treated as a sequence $Y = [y_1, y_2, ..., y_N]$ of independent acoustic feature vectors,
- $p(Y)$ is computed as a product of probabilities $p(y_t)$, $p(Y) = \prod_{t=1}^{T} p(y_t)$, and
- Each $p(y_t)$ is evaluated using a weighted sum of multivariate Gaussian PDFs defined on the acoustic feature space.

By analogy, in our 'segmental GMM':

- Y will be treated as a sequence of K independent segments $Y = \left[Y_1^{t_1}, Y_{t_1+1}^{t_2}, ..., Y_{t_{K-1}+1}^{N}\right]$ (where K depends on Y),

- $p(Y)$ is computed as a product of probabilities $p(Y_{t_{k-1}+1}^{t_k})$, $p(Y) = \prod_{k=1}^{K} p(Y_{t_{k-1}+1}^{t_k})$, where $t_0 = 0$ and $t_K = N$, and,
- Each $p(Y_{t_{k-1}+1}^{t_k})$ is evaluated using a trajectory-based segment model

Since the number of segment segment boundary points K and the values of the boundary points $t_1, t_2, ..., t_K$ are not known in advance, they must be calculated during the speaker-detection process using the segmental Viterbi decoder from section 3.3. By employing a segmental variant of the forward-backward algorithm for conventional HMMs, it would be possible to calculate $p(Y)$ by summing over all possible values of K and segmentations $t_1, t_2, ..., t_K$, and for an individual segment $[t_{k-1}+1, t_k]$ to calculate $p(Y_{t_{k-1}+1}^{t_k})$ by summing over all segment models. However, in the present study this was discounted on computational grounds, and also for the practical reason that it would necessitate substantial development of additional software within the 'SEGVit' toolkit. Instead we use the segmental Viterbi decoder to find the optimal value of K and segmentation $t_1, t_2, ..., t_K$, and for each segment $[t_{k-1}+1, t_k]$ we define $p(Y_{t_{k-1}+1}^{t_k}) = max_{\sigma} p(Y_{t_{k-1}+1}^{t_k}|\sigma)$, where σ ranges over all possible segment models.

In terms of a conventional GMM, this is analogous to computing the acoustic vector probability $p(y_t)$ by

$$p(y_t) = max_{m=1,...,M} p_m(y_t) \tag{11}$$

rather than by

$$p(y_t) = \sum_{m=1}^{M} p_m(y_t) \tag{12}$$

i.e. by choosing the best Gaussian component in the GMM instead of summing over all components. For consistency, and in order to focus on the 'frame-based' versus 'segment-based' comparison which is the subject of this research, we use equation (11) rather than (12) in all of our 'baseline' GMM experiments. Once this decision has been made, it will be seen that a conventional GMM is equivalent to a 'segmental GMM' in which the maximum segment duration τ_{max} is set to 1.

5.2 Construction of the 'segmental GMM'

Intuitively, the most natural approach to the problem of applying M-SHMMs to text-independent speaker verification is to replace the conventional GMM with a single segmental HMM. The 'segmental GMM' consists of M states, each associated with the type of variable-duration linear trajectory segment model described in section 3, specified by mean, slope and variance vectors in the acoustic space and a duration probability distribution. These states are configured in parallel, with a single initial, non-emitting, 'null' state and a single non-emitting

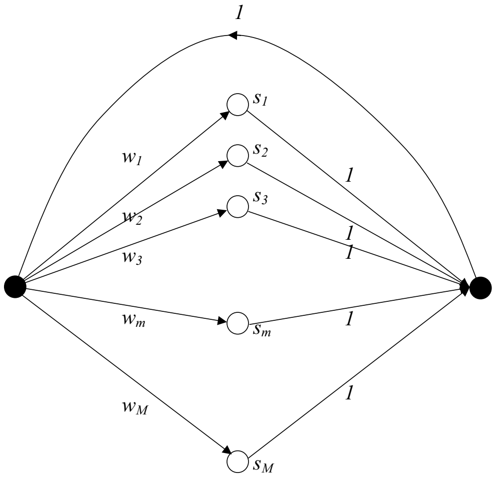


Figure 2: MSHMM structure for text-independent speaker verification.

final ‘null’ state (figure 2). The segmental states are analogous to the mixture components in a conventional GMM system, while the transition probability w_i from the initial null state to the i^{th} emitting segmental state corresponds to the GMM component ‘mixture’ weights.

Given a sequence $Y = [y_1, y_2, ..., y_N]$ which is claimed to correspond to an utterance spoken by speaker S, we compute the likelihood ratio:

$$L(S) = \frac{p(Y|S)}{p(Y)} \tag{13}$$

where the speaker-dependent probability $p(Y|S)$ is given by:

$$p(Y|S) = max_K max_{t_1,t_2,...,t_K} max_{\sigma^S_{i(1)},...,\sigma^S_{i(K)}} \prod_{k=1}^{K} (w^S_{i(k)}{}^{\lambda_1} p(Y^{t_k}_{t_{k-1}+1}|\sigma^S_{i(k)})\lambda_2) \tag{14}$$

In other words, for the speaker-dependent probability $p(Y|S)$ the maximum is taken over all possible numbers of segments K, all possible segmentations $t_1, t_2, ..., t_K$ of length K, and all possible sequences of length K $\sigma^S_{i(1)}, ..., \sigma^S_{i(K)}$ of states from the speaker-dependent model for speaker S. As before, λ_1 is the Language Model Scale Factor (LMSF) and λ_2 is the Token Insertion Penalty (TIP).

Similarly the BM probability $p(Y)$ is given by:

$$p(Y) = max_J max_{t_1,t_2,...,t_J} max_{\sigma^B_{i(1)},...,\sigma^B_{i(J)}} \prod_{j=1}^{J} ({w^B_{i(j)}}^{\lambda_1} p(Y^{t_j}_{t_{j-1}+1}|\sigma^B_{i(j)})\lambda_2) \quad (15)$$

For the background probability $p(Y)$ the maximum is taken over all possible numbers of segments J, all possible segmentations $t_1, t_2, ..., t_J$ of length J, and all possible sequences of length J $\sigma^B_{i(1)}, ..., \sigma^B_{i(J)}$ of states from the background model. We use different letters (K and J) for the segment sequence lengths in equations (14) and 15) to emphasise that, in general, both the number of segments and the segment indeces will be different for the speaker-dependent and background-model probability calculations.

5.2.1 The Language Model Scale Factor λ_1 and Token Insertion Penalty λ_2

The effect of the LMSF λ_1 is to control the influence of the individual 'mixture weights' w^S_j and w^B_j (in equation (15). A large value of λ_1 will 'sharpen' the distribution $\left[w^B_1, w^B_2, ..., w^B_M\right]$ and increase the influence of the weights. Conversely, if $\lambda_1 = 0$ then the weights will have no effect at all. The TIP λ_2 is a multiplicative penalty which is incurred each time a new segment is hypothesised. An explanation of a sequence Y which involves K segments will incur a penalty of ${\lambda_2}^K$. Thus setting $\lambda_2 = 1$ will have no effect, but setting $\lambda_2 > 1$ will favour longer sequences and setting $\lambda_2 < 1$ will favour shorter sequences.

In the 'SEGVit' M-SHMM toolkit, all probability calculations are done in the negative logarithmic domain (where maximising a probability is translated into minimising a cost), and parameters such as the LMSF and TIP are specified in the configuration file as values in that domain. In the negative logarithmic domain λ_1 becomes a multiplicative factor and λ_2 becomes an additive penalty. With respect to this domain, setting $\lambda_2 = 0$ will have no effect, but setting $\lambda_2 > 0$ will favour shorter segment sequences (and hence longer individual segments) and setting $\lambda_2 < 0$ will favour longer sequences (and hence shorter individual segments). Thus the TIP parameter λ_2 provides an external mechanism for influencing segment lengths.

5.3 Switchboard data sets used

The 2002 and 2003 NIST SRE subsets of Switchboard were obtained through NIST and LDC to enable us to evaluate the segmental GMM for speaker detection on the NIST 2003 SRE test. The experiments use:

- The one-speaker training material from the 2002 NIST SRE to train the BM,

- The one-speaker training data from the 2003 NIST SRE to train the SDMs, and
- A subset of approximately 50% of the one-speaker test data from the 2003 NIST SRE as test data.

An analysis of the systems used in the 2003 NIST SRE and the results obtained suggests that a suitable parameterisation of the speech signal would comprise mel frequency cepstral coefficients 1 to 18, plus energy, plus the corresponding Δ parameters. However, in the present system only the static parameters were used. This was partly to reduce the computational load, and partly because it was hoped that explicit modelling of speech dynamics would remove the need for the Δ parameters, as discussed earlier in section 4.1.1. The data was then parameterised as 18 mel frequency cepstral coefficients (MFCCs) plus an energy measure (C0) using the HTK 'HCopy' tool[3].

5.4 Training procedure for the 'segmental' GMM Background Model

An analysis of published results for conventional GMM systems suggests that an appropriate number of GMM components is of the order of 1024. However, some researchers (for example Auckenthaler and Mason) have reported good results on Switchboard data using as few as 500 components. In the case of our 'segmental GMM', the time taken to train and evaluate a model with 1024 segmental components would preclude an extensive investigation of the effect of different M-SHMM variants and parameters on speaker recognition performance. Hence, for the current experiments the number of segmental components in the 'segmental GMMs' was set to 300 ($M = 300$ in figure 2).

5.4.1 Factors influencing the performance of a 'segmental GMM'

The key parameters of the 'segmental GMM', whose effect on verification performance we want to measure, are as follows:

- The **maximum segment duration**. The parameter τ_{max} specifies the maximum allowable segment duration. If $\tau_{max} = 1$ then states are associated with individual feature vectors, and our 'segmental GMM' reduces to a type of conventional GMM. As τ_{max} increases, the model becomes 'more segmental' but the computational load increases. In our experiments on Switchboard, values of 1, 5 and 10 were chosen for τ_{max}.

[3] At first the MFCC-based parameterisation which uses an explicit measure of energy was chosen (MFC_E), however it was found that with this parameterisation HCopy gives incorrect results — abnormal huge positive or negative numbers — for some of the energy measure parameters of Switchboard data. This problem does not occur if the zeroth MFCC coefficient is used instead (MFC_0)

- The **trajectory slope**. This could be set to zero, estimated for the BM from training data and then maintained at this value for each speaker-dependent model, or reestimated for each speaker model. The significance of the trajectory slope parameters is likely to depend on the τ_{max} parameter: with slope being more significant for larger values of τ_{max}.

- The **segment duration model**. Again this could be trained from data for the BM and either passed unchanged to each speaker-dependent model or reestimated for each speaker-dependent model. Since duration is a segment-level, rather than frame-level, parameter, very few training examples of segment duration are likely to be contained in a typical speaker-dependent adaptation or training set. Therefore accurate estimation of a speaker-dependent duration model is likely to be an issue.

- The **language model control parameters λ_1 and λ_2**. As explained previously, the SEGVit system includes two parameters, LMSF (λ_1) and TIP (λ_2) (see section 5.2.1) which can be used to influence average segment duration. If λ_1 and λ_2 take their default values of 1 and 0, respectively (remember that these parameters operate in the negative log probability domain), then they have no effect on the Viterbi decoder. However, setting $\lambda_2 > 0$ will result in shorter state sequences and, hence, longer segments. Conversely, if $\lambda_2 < 0$ longer state sequences and shorter segments are preferred. Similarly, setting $\lambda_1 > 1$ will both sharpen the distribution of mixture weights (and therefore increase their influence) and decrease their magnitude (and therefore bias the decoder towards shorter state sequences and longer segments). Conversely, choosing $0 \leq \lambda_1 < 1$ will 'flatten' the distribution of mixture weights (and therefore reduce their influence) and increase their average value (and therefore bias the decoder towards longer state sequences and shorter segments). Thus by adjusting these two control parameters during training or testing, it is possible to influence the durational structures of the segments in the BM and speaker-dependent models.

5.4.2 'Segmental GMM' BM construction for Switchboard

As part of our previous research on TIMIT phone classification (Russell and Jackson 2005), we have developed software to produce sets of context-sensitive triphone M-SHMMs of varying sizes (using the monophone and biphone 'backoff' approach described earlier). Using this software we have developed TIMIT-based model sets with between 104 and 5,989 models (or, equivalently, between 312 and 17,967 states). By combining all or a subset of the states of a suitable family of models into a single, integrated M-SHMM of the type depicted in figure 2 we hoped that we could obtain a suitable initial model to 'seed' Viterbi reestimation of our segmental BM for Switchboard. Estimation of the target speaker models could then proceed as previously described. For this pilot experiment we chose the maximum segment duration τ_{max} to be equal to 5.

Unfortunately this did not prove to be the case. The dissimilarity between the TIMIT-based models and the Switchboard data was such that nearly 80% of the MSHMM states were not visited at all during reestimation. After two iterations of the reestimation, only 20% of the MSHMM states had non-zero 'occupancy' and could therefore be reestimated. Thus the effective number of states was significantly reduced. We concluded that it is not possible to use segmental states estimated estimated on TIMIT as initial models for work on Switchboard.

As an alternative we estimated the mean values of 300 segments using k-means clustering applied to a randomly chosen subset of the Switchboard 2002 data. The initial segment trajectory slope values were set to zero and the state duration distributions were set to be uniform.

These initial segment models were used to construct an initial 'segmental GMM' Background Model, which was optimised using the Viterbi-based M-SHMM reestimation functions in the 'SEGVit' software toolkit and the NIST 2002 SRE one-speaker training set. The segment trajectory means and variances were reestimated first, using 4 iterations of Viterbi training. Then the segment trajectory means, slopes and variances were reestimated for a further 5 iterations. The duration probabilities were only reestimated in the final, 5th, iteration.

Different maximum segment lengths corresponding to $\tau_{max} = 1, 5$ and 10 were chosen to make 3 sets of models, which we refer to as $SW1$, $SW5$, and $SW10$. These models were built to test the effect of maximum segment duration on speaker-detection performance. For all model sets except $SW1$, the segment trajectory means and slopes, variances and the segment duration distributions were estimated. In the case of $SW1$, only the segment trajectory means and variances were reestimated, the trajectory slopes were set to 0 and the duration length can only be 1. The model $SW1$ was treated as the counterpart of the traditional GMM system and used as our baseline system.

5.5 Training procedure for the speaker-dependent 'segmental GMMs'

Each trained BM from section 5.4.2 was used to reestimate a speaker-dependent 'segmental GMM' (SDM) for each of the test speakers in the 2003 Switchboard test set. Data from the 2003 Switchboard training set was used to reestimate these models.

For the BM set $SW5$ ($\tau_{max} = 5$), three different sets of SDMs were produced:

- In the first set, $SW5\_0$, the segment trajectory mean vectors were reestimated but the slope vectors were set to zero in both the BM and SDMs.

- In the second set, $SW5\_1$, only the segment trajectory mean values were reestimated. The segment trajectory slopes in these models are therefore the same as those of the corresponding segment models in the BM.

- In the third set, $SW5\_2$, the segment trajectory slopes were also reestimated, along with the segment trajectory means.

For the speaker-dependent models the segment duration models and variance parameters were not reestimated because of the limited amount of training data which is available for each speaker. The trajectory means were reestimated in all cases.

The effects on performance of setting the trajectory slope values to zero in both the BM and SDMs, reestimating them for the BM but not the SDMs, or reestimating them for the BM and SDMs, were tested experimentally.

5.6 Initial experiments on the NIST 2003 single-speaker evaluation set

Because of the need to run segmental Viterbi decoding and to compute segment-level probabilities, the computational load associated with our 'segmental GMM' is significantly greater than that associated with a conventional GMM. In order to reduce this computational cost and to improve experimental turn-around time, speaker-detection experiments were conducted using just half of the male test speakers (671 speakers) and half of the female test speakers (1042 speakers) from the NIST 2003 single-speaker evaluation set.

As specified in the NIST 2003 evaluation documentation, for each test file, 11 different verification tests were performed. This in turn involved 12 probability calculations - one for the background model and 11 for the speaker-dependent models.

The following experiments were conducted:

- **Experiment 1**: This experiment investigated the effects on performance of setting the trajectory slope values to zero in both the BM and SDMs ($SW5\_1$), reestimating the trajectory slope vector for the BM but not for the SDMs (so that the SDM trajectory slope vectors are equal to the corresponding BM slope vectors, $SW5\_2$), and reestimating the slope vector for both the SDMs and the BM ($SW5\_3$). In this experiment $\tau_{max} = 5$.

- **Experiment 2**. The performances of the systems with maximum duration τ_{max} set to 1 ($SW1$), 5 ($SW5$) and 10 ($SW10$) were compared. In these experiments all of the BM trajectory parameters were reestimated and used to seed the corresponding SDM parameters, and all of the SDM parameters were then reestimated (except in the case of $SW1$, where the slope vectors are all zero - this is the baseline system)

5.7 Speeding up experiment turn-around time

It has already been noted that the computational load associated with M-SHMMs is an important issue (see section 3.3 and, in particular the discussion after equation (3.3)).

For this reason, the time taken to train the BM and the even longer time required for testing meant that it would not be possible to evaluate many different variations of the 'segmental GMM' system. However, as we have no previous experience of applying these models to Switchboard, many experiments need to be conducted to derive optimal values for the maximum segment durations and to establish the utility of the different trajectory parameters.

It has already been noted that only half of the test set from the NIST 2003 was used, and this reduces the computation in testing by 50%. The number of segmental states in the model was also kept low at 300. However, the computational load was still prohibitive.

5.7.1 Parallelisation of the SEGVit toolkit

The 'SEGVit' software toolkit has been modified so that model training can be conducted in parallel on a 'grid' of computers. However the computation time is still prohibitively long for a large detection task. For example, we estimate that an evaluation of our reduced system, with $\tau_{max} = 15$ will take between 20 and 25 days on our 6-node cluster.

5.7.2 Beam pruning and duration pruning

Techniques which work for recognition, such as Beam Pruning have been extended to the 'SEGVit' toolkit during the period of this project, but they are much less effective for speaker detection than for speech recognition. This is because at present there is effectively no syntax to constrain possible segment sequences. In other words, because each segment in the 'segmental GMM' can be preceded by every other segment, pruning out paths in the past does not alter the number of segments which have to evaluated in the present. We also developed a new technique which we refer to as 'Duration Pruning' whereby a segment probability is not calculated if the probability of its duration is below a pre-determined threshold. Again, this technique works well for phone recognition experiments on TIMIT but appears to be less useful for speaker detection experiments on Switchboard.

5.7.3 Auckenthaler's method for reducing computational load

In a further attempt to speed up our experiments, we investigated a technique described by Auckenthaler in his thesis (Auckenthaler 2001).

Auckenthaler proposes two methods to reduce the computational load in a conventional GMM-based speaker detection system:

- In Auckenthaler's first method a 'bigram' grammar for sequences of GMM mixture components is built using mixture component sequences observed on the training data. For each mixture component m, this bigram grammar is used to identify the sub-set of N components which are most probable at time $t+1$ if the m^{th} component is most probable at time t. During recognition the decoder is constrained so that if a particular mixture component m is used at time t, then only this pre-determined subset of mixture components is considered at time $t+1$.

- The second method exploits the link between the BM and each of the SDMs. Since each SDM is seeded by the BM, it is argued that there is a strong connection between the m^{th} component of the BM and the corresponding m^{th} component of the SDM. Auckenthaler reasoned that if this is the case, then given a test utterance $Y = [y_1, ..., y_T]$ the sequence of mixture components $m_{i(1)}, ..., m_{i(T)}$ which is optimal for the BM should be close to optimal for each SDM. Therefore, once the optimal sequence of components has been computed for the BM, Auckenthaler uses exactly the same sequence for each of the SDMs. In essence, this means that for each acoustic vector y_t only the probability $b_{m_{i(t)}}(y_t)$ needs to be evaluated and the remaining $M-1$ probabilities b_m need not be evaluated. For a 500 component GMM, this means a 499-fold reduction in computational load.

In (Auckenthaler 2001) the effects of these techniques on detection performance are documented.

We developed new software within the 'SEGVit' system to implement our analogy to Auckenthaler's second scheme. First the optimal state sequence between a given test utterance and the BM was computed. We then assumed that the same state sequence is valid for the SDMs, thereby removing the need to do further Viterbi decoding. We tested this method on the system with $\tau_{max} = 10$. The new method effectively reduced the processing time for testing from more than two weeks to within 3 days, with little loss in system performance. For a system with $\tau_{max} = 15$, the verification process can be completed within 5 days, and the time taken for whole training and test process decreases from about one month to only 7 or 8 days.

5.8 Effect of λ_1 and λ_2 on segment duration

As described earlier, the Language Model Scaling Factor (λ_1) and Token Insertion Penalty (λ_2) can be used to alter the statistics of segment duration. Figure 3 shows the effect of varying the second parameter, λ_2 on segment duration statistics. In these experiments λ_1 was set to 1 while λ_2 was varied between -10 and 100. It is important to note that these statistics are obtained from the test data. The BM and SDMs were trained with $\tau_{max} = 10$, $\lambda_1 = 1$ and $\lambda_2 = 0$. Figure 3 shows that the average segment duration for the 'default' case

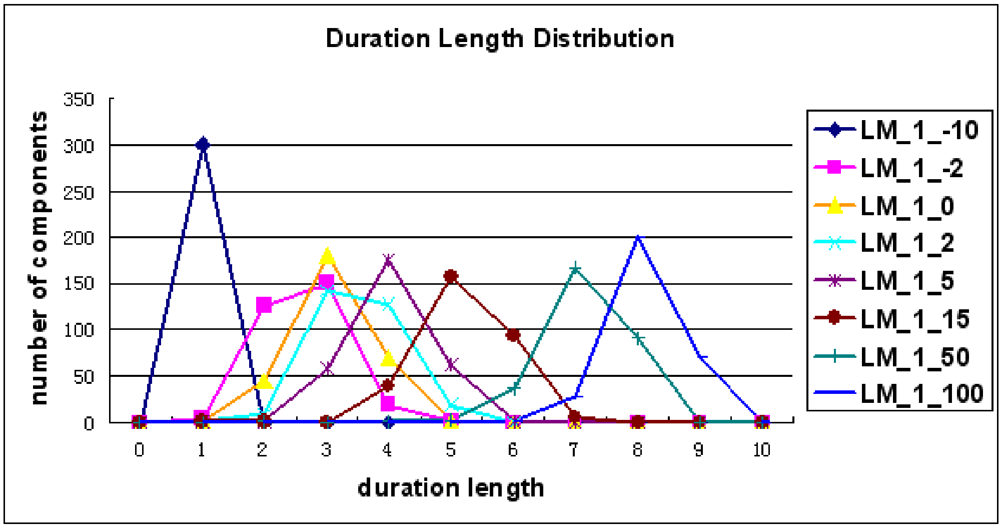


Figure 3: *Duration length distributions for different values of the Token Insertion Penalty* λ_2. *LM_x_y refers to the case where* $\lambda_1 = x$ *and* $\lambda_2 = y$. *LM_1_0 is the default.*

where $\lambda_1 = 1$ and $\lambda_2 = 0$ is 30ms, with a minimum duration of 5ms and a maximum duration of 70ms. By increasing λ_2 to 10 the most probable duration is increased to 80ms. For such large values of λ_2 it is likely that there is a conflict between the effect of λ_2, which is to increase the expected segment duration, and the hard upper-bound on segment duration imposed by τ_{max}. Setting $\lambda_2 = -2$ shifts the duration distribution slightly to the left (towards shorter durations), while setting $\lambda_2 = -10$ causes all segments to have minimum duration, which is 10ms (or one acoustic vector).

5.9 Results of Switchboard experiments

5.9.1 Effect of the trajectory slope vector

The results for the first experiment (**experiment 1**), with model sets $SW5\_1$ $SW5\_2$ and $SW5\_3$[4] are shown as DET curves in Appendix A (figure 5). Recall that $\tau_{max} = 5$ in these experiments. The figure shows that the equal error rate for all three systems is approximately 14%. The best performance is obtained using speaker-dependent trajectory slopes (scheme 3), but the difference between this and the other results is very small and unlikely to be significant. The experiment in which non-zero BM slopes are estimated, and used to seed the speaker-model slopes but are not subsequently reestimated (scheme 2), gives results which are almost indistinguishable from the zero-slope result (scheme1).

[4] Recall that conditions 1, 2 and 3 correspond to trajectory slopes set to zero in the BM and SDMs; BM trajectory slopes learnt but not reestimated in the SDMs; BM trajectory slopes learnt and reestimated for the SDMs

5.9.2 Effect of the maximum segment duration τ_{max}

The results of the second experiment (**experiment 2**), for systems with different maximum durations, namely $SW1$ ($\tau_{max} = 1$), $SW5$ ($\tau_{max} = 5$) and $SW10$ ($\tau_{max} = 10$) are shown in Appendix A (figure 6). The figure shows that the systems with $\tau_{max} = 5$ and $\tau_{max} = 10$ work very slightly better than the system with $\tau_{max} = 1$, but the differences are too small to be significant. Recall that $SW1$ is our approximation to a conventional GMM.

These results are certainly not as we expected. We expected that in experiment 1 scheme 1 would give poorer results than schemes 2 and 3, and thereby demonstrate the utility of modelling dynamics by incorporating a non-zero slope parameter. In fact this experiment provides little evidence to support the hypothesis that the use of linear segment models with non-zero trajectory slopes is beneficial for speaker detection. This result contrasts with the previous result for YOHO, where there does appear to be a benefit.

In the second set of experiments we expected that $SW10$, with maximum segment duration set to 10, would outperform $SW5$ ($\tau_{max} = 5$), and that $SW5$ would in turn outperform $SW1$ ($\tau_{max} = 1$). However there is little evidence in the results to support this expectation. It should also be noted that the results of experiments 1 and 2 are consistent. If (as suggested by the results of experiment 1) there is no benefit from using a model based on 'dynamic' trajectories with non-zero slope, then one would not expect to observe any benefit from longer segments (since a long, constant segment can be modelled just as well by a sequence of short, constant segments).

We note that all of these results are clearly much worse than the best performance obtained on the full 2003 test set using a conventional GMM system, which is a little over 5% equal error rate. This was obtained using a 2048 component GMM system, T-norm and a biologically inspired acoustic parameterisation. However, the goal of these initial experiments was not to challenge the state-of-the-art in terms of performance, but to conduct comparative experiments to determine the benefits of using a dynamic, trajectory-based model.

5.9.3 Effects of reducing the computational load

The results obtained by applying the 'segmental GMM' version of Auckenthaler's second method, described in section 5.7.3, are shown in the DET curves in Appendix B. Each figure shows two DET curves. The dashed (blue) line is the same in all of the figures and is included as a baseline. It shows the DET curve obtained when separate Viterbi decoding is applied to each of the SDMs (i.e. Auckenthaler's method is not used). For these experiments $\lambda_1 = 1$, $\lambda_2 = 0$ and $\tau_{max} = 10$.

The solid (red) DET curves show the results of applying Auckenthaler's method (i.e. using the optimal state sequence obtained using Viterbi decoding relative to the BM to calculate the SDM probabilities) together with different values of language model control parameters λ_1 and λ_2 ($\lambda_1 = 1; \lambda_2 \in \{-10, -2, 0, 2, 5, 15, 50, 100\}$).

Figure 7 shows a direct comparison, for $\lambda_1 = 1$ and $\lambda_2 = 0$, of the results obtained with and without the computational reduction due to Auckenthaler's method. The figure shows that the DET curves are almost identical, with the reduced computation method showing small gains at each extreme of the DET curve but performinslightly worse towards the centre of the curve. We conclude that the large reduction in computational load which results from using the optimal BM state sequence to calculate the SDM probabilities is not compromised by a significant change in speaker detection performance.

Turning now to the effects of varying the Token Insertion Penalty λ_2 (figures 8 to 14) we see that there is very little difference between the DET curves for the different values of λ_2, despite the large variation in expected segment duration shown in figure 3. In particular, it is certainly not the case that (as one might have expected) performance reaches a maximum for some positive value of λ_2. Indeed, larger values of λ_2 lead to decreases in performance, and the best performance is obtained with $\lambda_2 = -2$. From figure 3 this value of λ_2 corresponds to an expected segment duration of between 20ms and 30ms. It seems that shorter segment duration lengths give the best performance, which is quite different from what we expected but consistent with the results for varying τ_{max}.

At this point we noted a possible incompatibility in these experiments. The language model control parameter λ_2 was only varied during testing and not during training. Therefore it's effect on segment duration during testing is incompatible with the duration models learnt during training. To make the effect of the language model control parameters compatible with the model durations, additional experiments were carried out. In these experiments, the language model control parameter λ_2 was the same in model training as in testing ($\lambda_2 \in \{5, 15, 50\}$).

The results of these experiments are shown in Appendic C. The DET curves for the systems which use the optimal BM state sequences when calculating SDM probabilities are shown with a solid (green) grey line (this is the 'Auckenthaler method'). The DET curves for systems which apply Viterbi decoding separately to the BM and SDMs are shown with a dashed (blue) line ($\lambda_1 = 1$; $\lambda_2 = 0$). The DET curve for a conventional GMM system is shown with a solid, black line.

The results are similar to those in Appendix B. These support the hypothesis that the results in Appendix B are not affected significantly by use of different values of λ_2 in training and testing. As in Appendix B, the DET curves in Appendix C show a trend whereby performance decreases as λ_2 (and hence the average segment durations) increases. The figures confirm, again, that Auckenthaler's method has little effect on performance.

6 Visualisation of the ‘segmental GMM’ segment models

The results of our speaker detection experiments on Switchboard are not as expected. We have been unable to demonstrate any benefit from the use of ‘dynamic’ segments based on linear trajectories with non-zero slope. Hence we have also not been able to demonstrate any benefit from the use of longer segments. This results is at odds with our earlier speaker detection results on YOHO, described in section 4.2, and with the phone recognition results presented in (Russell and Jackson 2005).

In order to try to understand this result, we have written a MatLab program to visualise the individual segment models in the ‘segmental GMM’. The results are illustrated in Appendix D.

For each segment, we computed linear trajectories for all 19 MFC coefficients. The length of a segment is its average length, based on its duration distribution. This results in a sequence of 19 dimensional MFCC vectors. We then applied an inverse Discrete Cosine Transform to each of these vectors to obtain a mel frequency spectrum, whose frequency axis was then warped to obtain a linear frequency spectrum. The resulting sequence of linear spectral vectors is displayed as a grey-scale spectrogram to give one of the figures in appendix C.

Visual inspection of these ‘spectrograms’ suggests that they are all valid speech segments, and that they correspond to different components of a plausible segmental model of speech. For example, the second segment in the third row on the first page of appendix D is clearly vowel like, while the segment in position (5,1) is more fricative-like. The figures show a mixture of stationary and non-stationary segments.

In summar, visual inspection of these segments does not reveal any obvious problems, and a method for more detailed analysis is needed.

7 Provision of SEGVit software toolkit to AFRL

In addition to conducting the speaker-detection experiments which are described in this report, we also provided Dr Timothy Anderson’s research group at the Air Force Research Laboratory (AFRL) at Wright-Patterson Air Force Base, Dayton, Ohio, with a copy of the SEGVit toolkit and with guidance on how to use it. This was to enable AFRL to evaluate M-SHMMs for phone-based speaker-detection on Switchboard, using the SRI phone-level annotations of the Switchboard corpus. To achieve this, various changes to the SEGVit software were required, and these were implemented, tested and sent to AFRL.

8 Conclusions and further work

This report has described the main results obtained during the 12 month EOARD project #033060 “Speaker verification using a dynamic, ‘articulatory’ segmental hidden Markov model”, which started on 1st October 2003.

The results of text-dependent speaker verification experiments on the YOHO corpus are presented first. These show a 44% decrease in false acceptance rate for a segmental HMM based system, relative to a conventional HMM-based system, for the same false rejection rate. However, the false rejection rate is too small to draw firm conclusions about the relative merits of the two approaches. Hence our attention moved away from YOHO to the Switchboard corpus.

To conduct text-independent speaker detection experiments on Switchboard we developed a type of ‘segmental GMM’, which models speech as a sequence of outputs of a set of linear-trajectory-based statistical segment models. However, due to the requirement to do segmental Viterbi decoding and the need to compute segment-level probabilities, the computational demands of this model are prohibitive. We overcame this problem as follows:

- We only conducted experiments on 50% of the NIST 2003 SRE single-speaker test set
- We developed a parallel version of the ‘SEGVit’ software toolkit, which enabled training to be spread over a grid of computers
- We incorporated versions of beam pruning and ‘duration pruning’ into the ‘SEGVit’ software.
- We successfully extended Auckenthaler’s method, whereby the optimal BM state sequence is used to compute the SDM probabilities, to our ‘segmental GMM’

By combining these methods we were able to run a speaker detection experiment on our reduced NIST 2003 test set in a few days. For example, the time taken to evaluate a system with maximum segment duration equal to 10 on our 6 node cluster was reduced from two weeks to three days.

The main results of our experiments on Switchboard are as follows:

- The techniques described above to reduce the computational load were very successful and had no significant effect on speaker detection performance
- On the Switchboard corpus, we were unable to demonstrate any benefit from the use of ‘dynamic’ segments based on linear trajectories with non-zero slope
- Consequently, we were unable to demonstrate any significant benefit from the use of long segments. Indeed, the best performance was obtained with segments with an expected duration of between 20ms and 30ms, obtained by setting the Token Insertion Penalty λ_2 to -2.

The discrepancy between the performance of M-SHMMs for text-dependent detection on YOHO and their performance for text-independent detection on Switchboard is puzzling. There are at least two possible explanations:

- The experiments on YOHO are text-dependent and use the YOHO word-level labeling. This labeling enabled us to use phone-level models in speaker detection. By contrast, no labels were used in the case of Switchboard and the models were 'machine learnt' segment models with no explicit phonetic interpretation. It could be that some sort of explicit labeling is needed to guide the segmental model building process. However, parallel experiments were conducted at AFRL using the 'SEGVit' software toolkit and the automatically-derived SRI Switchboard phone-level labels to build phone-level trajectory-based M-SHMMs. These models performed worse than a conventional GMM-based system in tests where both systems had comparable numbers of parameters. This suggests that the absence of phone level labeling in Switchboard is not the answer.
- An alternative explanation is that the discrepancy is due to the different styles of speech in the YOHO and Switchboard corpora. While YOHO contains recordings of read speech, Switchboard comprises recordings of conversational speech over various telephone chanels. The poorer quality of the Switchboard speech might have caused difficulty for the data-driven segment model learning process, or, alternatively, cues which the segment models were able to use in the YOHO corpus may be absent in Switchboard.

To test the hypothesis that the poor quality of the Switchboard data compromises the data-driven segment model learning process, we developed MatLab code to visualise the individual segment models. However, inspection of these representations of the individual segment models in our 'segmental GMM' does not reveal any obvious problems - the segment model set appears to cover a range of speech-like segments. However, the true quality of the segment models is difficult to judge, and better visualisation tools are needed. In particular we need to extend our segment model visualisation tools to enable us to display real spectrograms of Switchboard data alongside a representation of the spectrogram corresponding to the optimal sequence of segment models. This should give a much better understanding of the accuracy of the model.

At present, our main conclusion is that the fact that the inclusion of dynamic segments, corresponding to trajectories with non-zero slope, consistently fails to improve speaker detection accuracy on Switchboard, suggests that these dynamic regions do not contain information which helps to differentiate between speakers in this corpus. If this is true, it would go some way to explaining the success of conventional GMM-based approaches to speaker detection on Switchboard. It is also possible that these dynamic regions are more useful in a non-conversational corpus like YOHO.

To confirm this hypothesis, we believe that it is important to conduct further work to determine the exact contribution of dynamic regions of a speech signal to speaker-detection accuracy. In the context of our current work, we can define dynamic regions of a speech signal to be those which align with segments with large slope values in the segmental GMM. By measuing the contribution to the likelihood ratio $\frac{p(Y|S)}{p(Y)}$ of individual segments of a speech signal, we will be able to measure the relative contributions of static and dynamic segments to the speaker-detection decision. We propose to apply this analysis to Switchboard, to test our hypothesis, and to YOHO to see if the contribution of dynamic regions is more important for a read, and therefore more carefully articulated, corpus.

References

Auckenthaler, R. (2001). *Text-independent speaker verification with limited resources.* Ph. D. thesis, University of Wales Swansea.

Auckenthaler, R., M. J. Carey, and H. Lloyd-Thomas (2000, January/April/July). Score normalisation for text-independent speaker verification systems. *Digital Signal Processing 10*(1-3), 42–54.

Bridle, J. S. (2004). Towards better understanding of the model implied by the use of dynamic features in hmms. In *Proc. Int. Conf. on Spoken Lang. Proc.,* Jeju Island, Korea.

Bridle, J. S., M. D. Brown, and R. M. Chamberlain (1983). Continuous connected word recognition using whole-word templates. *Radio Engineer 53*, 167–177.

Deng, L. and J. Ma (2000). Spontaneous speech recognition using a statistical coarticulatory model for the vocal-tract-resonance dynamics. *J. Acoust. Soc. Am. 108*(6), 3036–3048.

Ferguson, J. D. (1980). Hidden markov analysis. *in Hidden Markov Models for Speech, Institute for Defense Analysis, Princeton, NJ*.

Garofolo et al., J. S. (1993). *TIMIT Acoustic-Phonetic Continuous Speech Corpus.* Univ. Pennsylvania, Philadelphia, PA: Linguistic Data Consortium.

Gauvain, J.-L. and C. Lee (1994). Maximum a-posteriori estimation for multivariate gaussian mixture observations of markov chains. *2*, 291–298.

Higgins, A. (1990). Yoho speaker verification. In *Presented at the Speech Research Symposium, Baltimore, MD.*

Holmes, J. and W. Holmes (2001). *Speech synthesis and recognition* (2nd ed.). London and New York: Taylor and Francis.

Holmes, J. N. (1973). The influence of glottal waveform on the naturalness of speech from a parallel formant synthesiser. *IEEE Transactions on Audio and Electroacoustics AU-21*, 298–305.

Holmes, J. N., I. G. Mattingly, and J. N. Shearme (1964). Speech synthesis by rule. *Language & Speech 7*, 127–143.

Holmes, W. J. and M. J. Russell (1999). Probablistic-trajectory segmental HMMs. *Comp. Speech & Lang. 13*(1), 3–37.

Leggetter, C. and P. C. Woodland (1995). Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *9*(2), 171–185.

Liu, Y., M. J. Russell, and M. J. Carey (2004). Speaker recognition using a trajectory-based segmental hmm. In *Proc. Odyssey'04, The Speaker and Language Recognition Workshop* Toledo, Spain, pp. 45–50.

Ostendorf, M., V. V. Digalakis, and O. A. Kimball (1996). From HMM's to segmental models: a unified view of stochastic modeling for speech recognition. *IEEE Trans. on Spch. & Aud. Proc. 4*(5), 360–378.

Reynolds, D. A. (1992). *A Gaussian mixture modeling approach to text independent speaker identification.* Ph. D. thesis, Georgia Institute of Technology.

Richards, H. B. and J. S. Bridle (1999). The HDM: a segmental Hidden Dynamic Model of coarticulation. In *Proc. IEEE-ICASSP,* Phoenix, AZ, pp. 357–360.

Russell, M. J. and P. J. B. Jackson (2005). A multiple-level linear/linear segmental HMM with an 'articulatory' intermediate layer. *to appear in Comp. Speech & Lang.*.

Young, S. J., J. Odell, D. Ollason, V. Valtchev, and P. Woodland (1997). *The HTK Book* (v2.1 ed.). Cambridge, UK: Entropic Camb. Res. Lab.

APPENDIX A

Results of text-dependent and text-independent speaker verification experiments.

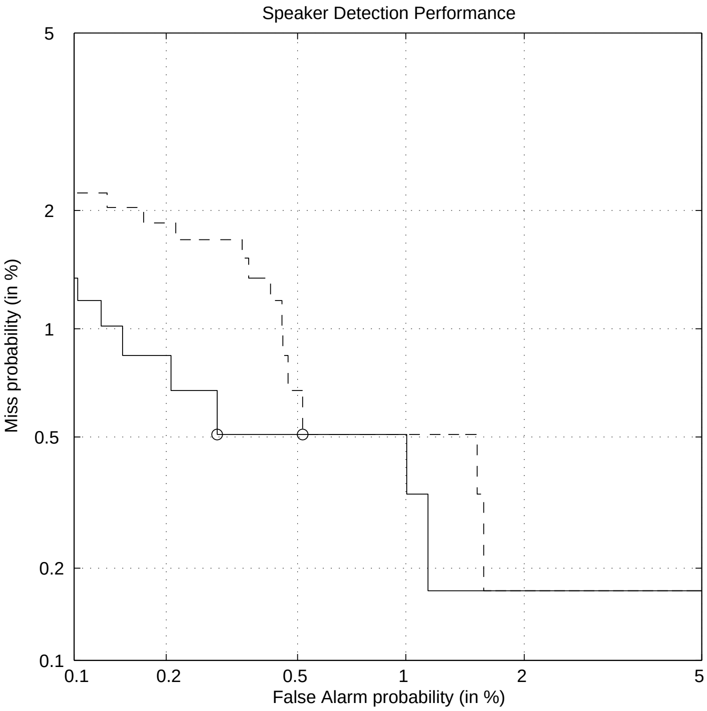


Figure 4: Text-dependent speaker verification results on YOHO using HMMs (dashed line) and MSHMMs (solid line).

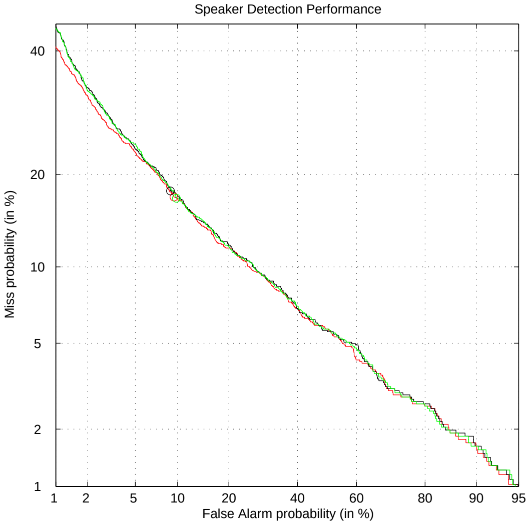


Figure 5: Speaker verification results on a 50% subset of the NIST 2003 Switchboard 'one-speaker' test set, using linear trajectory segmental HMMs with $\tau_{max} = 5$. The results are for trajectory slopes set to zero in both the BM and the SDMs (scheme 1 - black line), trajectory slopes reestimated for the BM but not reestimated for the SDMs (scheme 2 - green line), and reestimated for each of the SDMs (scheme 3 - red line).

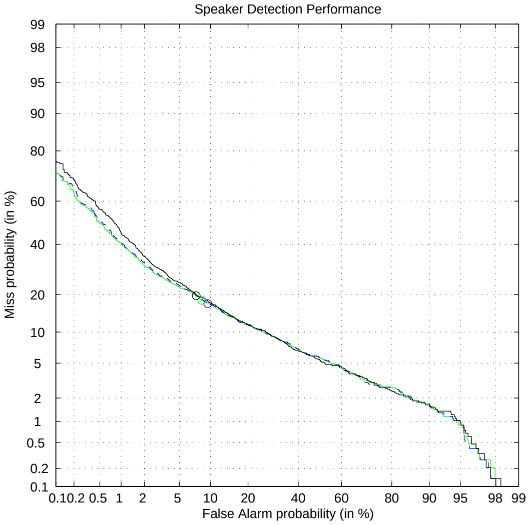


Figure 6: Speaker verification results on a 50% subset of the NIST 2003 Switchboard 'one-speaker' test set, using linear trajectory segmental HMMs with $\tau_{max} = 1$ (scheme 1 - black solid line), $\tau_{max} = 5$ (scheme 2 - dashed line), $\tau_{max} = 10$ (scheme 3 - green solid line) .

APPENDIX B

Results of experiments to investigate the effect of using the BM optimal state sequence when computing the SDM probabilities, and different values of λ_1 and λ_2.

The DET curves for the systems which use the optimal BM state sequences when calculating SDM probabilities are shown with a solid line (this is the ‘Auckenthaler method’). The DET curves for systems which apply Viterbi decoding separately to the BM and SDMs are shown with a dashed line. This line is the same in all of the figures and corresponds to $\lambda_1 = 1$; $\lambda_2 = 0$.

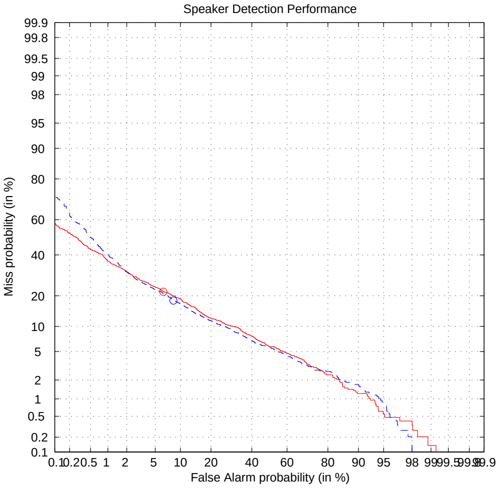


Figure 7: $\lambda_1 = 1$; $\lambda_2 = 0$.

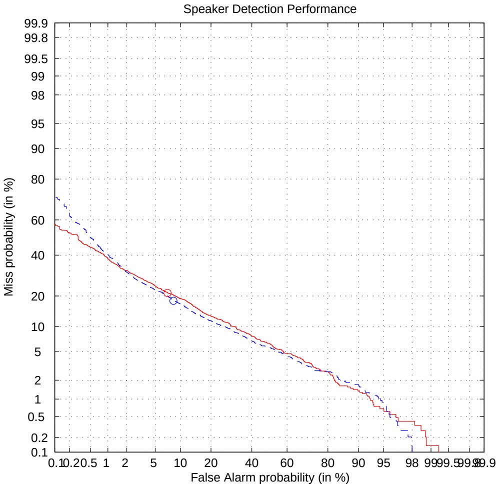


Figure 8: $\lambda_1 = 1$; $\lambda_2 = 2$.

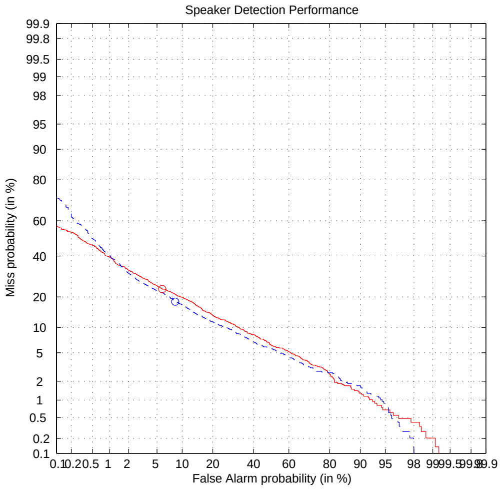


Figure 9: $\lambda_1 = 1; \lambda_2 = 5$.

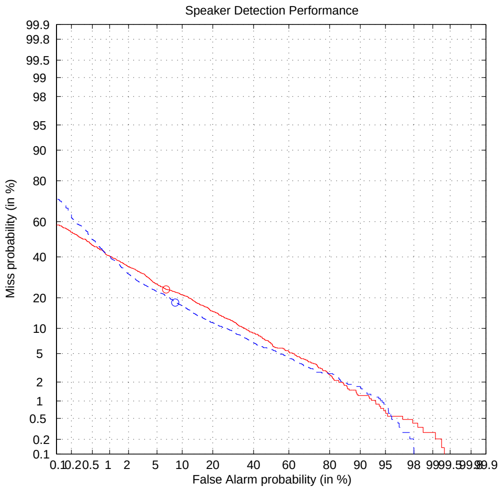


Figure 10: $\lambda_1 = 1; \lambda_2 = 15$.

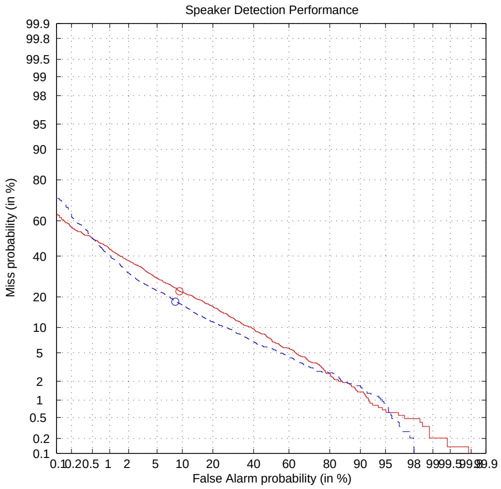


Figure 11: $\lambda_1 = 1; \lambda_2 = 50$.

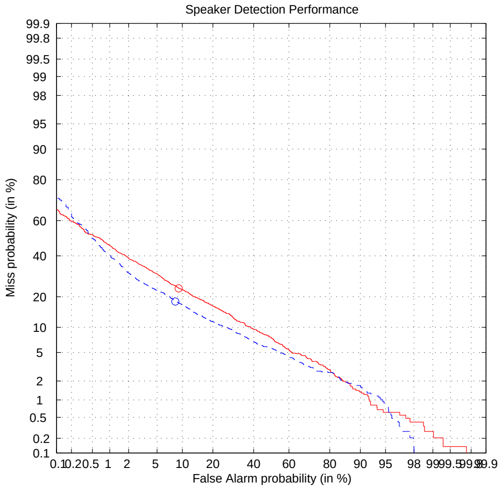


Figure 12: $\lambda_1 = 1; \lambda_2 = 100$.

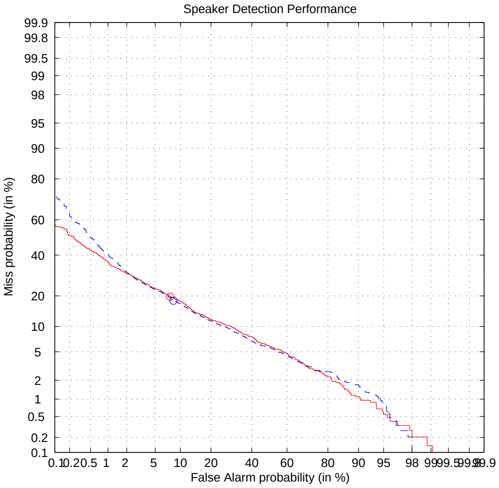


Figure 13: $\lambda_1 = 1; \lambda_2 = -2$.

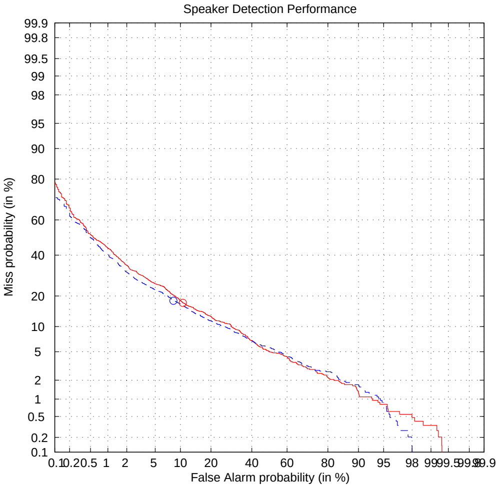


Figure 14: $\lambda_1 = 1; \lambda_2 = -10$.

APPENDIX C

Results of experiments to investigate the effect of using the BM optimal state sequence when computing the SDM probabilities, and different values of λ_1 and λ_2. Experiments are as in Appendix B, except that the same values of λ_1 and λ_2 are used in training and recognition.

The DET curves for the systems which use the optimal BM state sequences when calculating SDM probabilities are shown with a solid grey line (this is the 'Auckenthaler method'). The DET curves for systems which apply Viterbi decoding separately to the BM and SDMs are shown with a dashed line ($\lambda_1 = 1$; $\lambda_2 = 0$). The DET curve for a conventional GMM system is shown with a solid, black line.

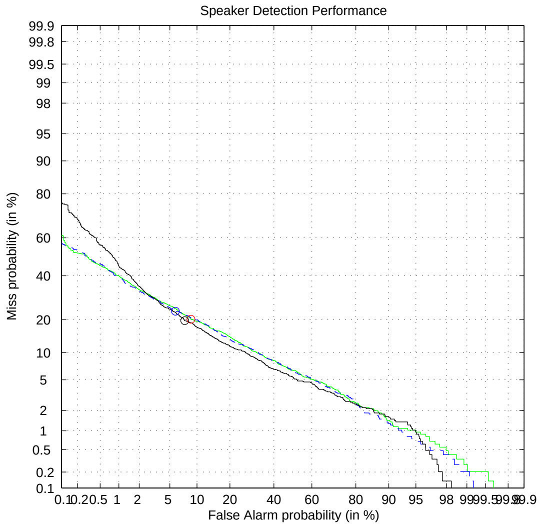


Figure 15: LMScale = 1; LMInsP = 5 .

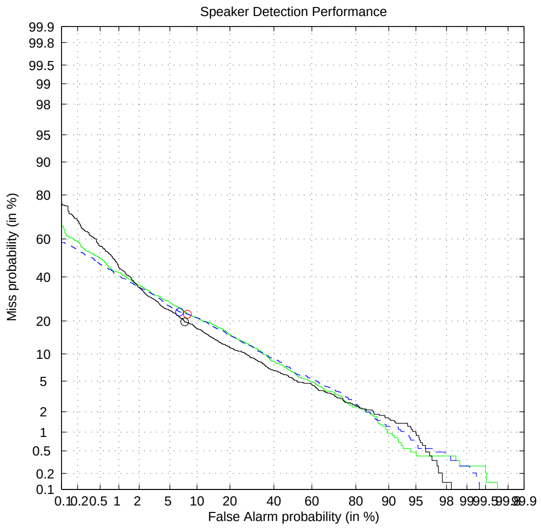


Figure 16: LMScale = 1; LMInsP = 15 .

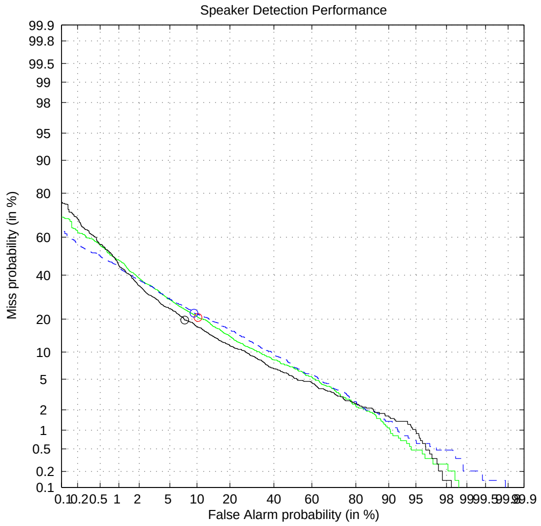
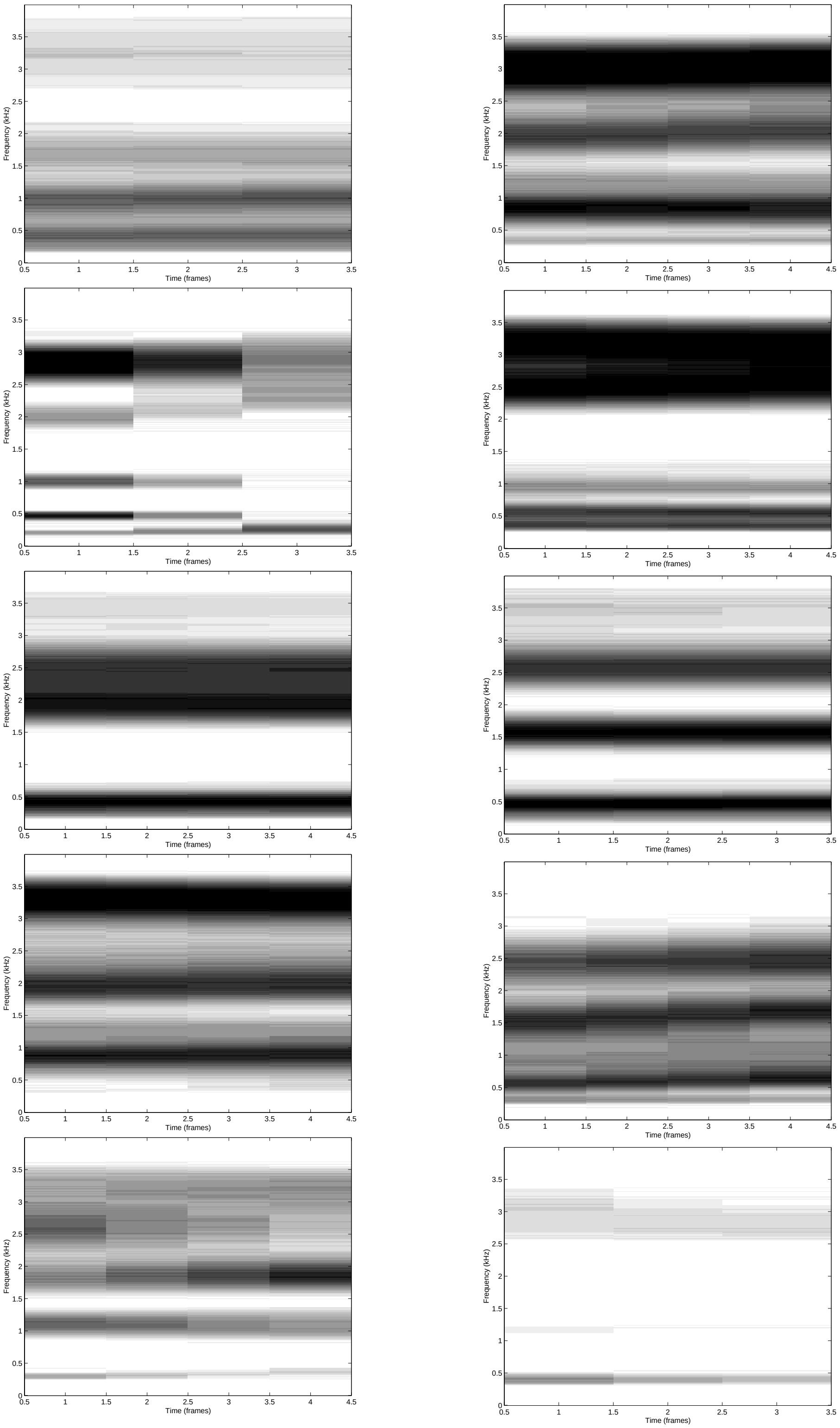
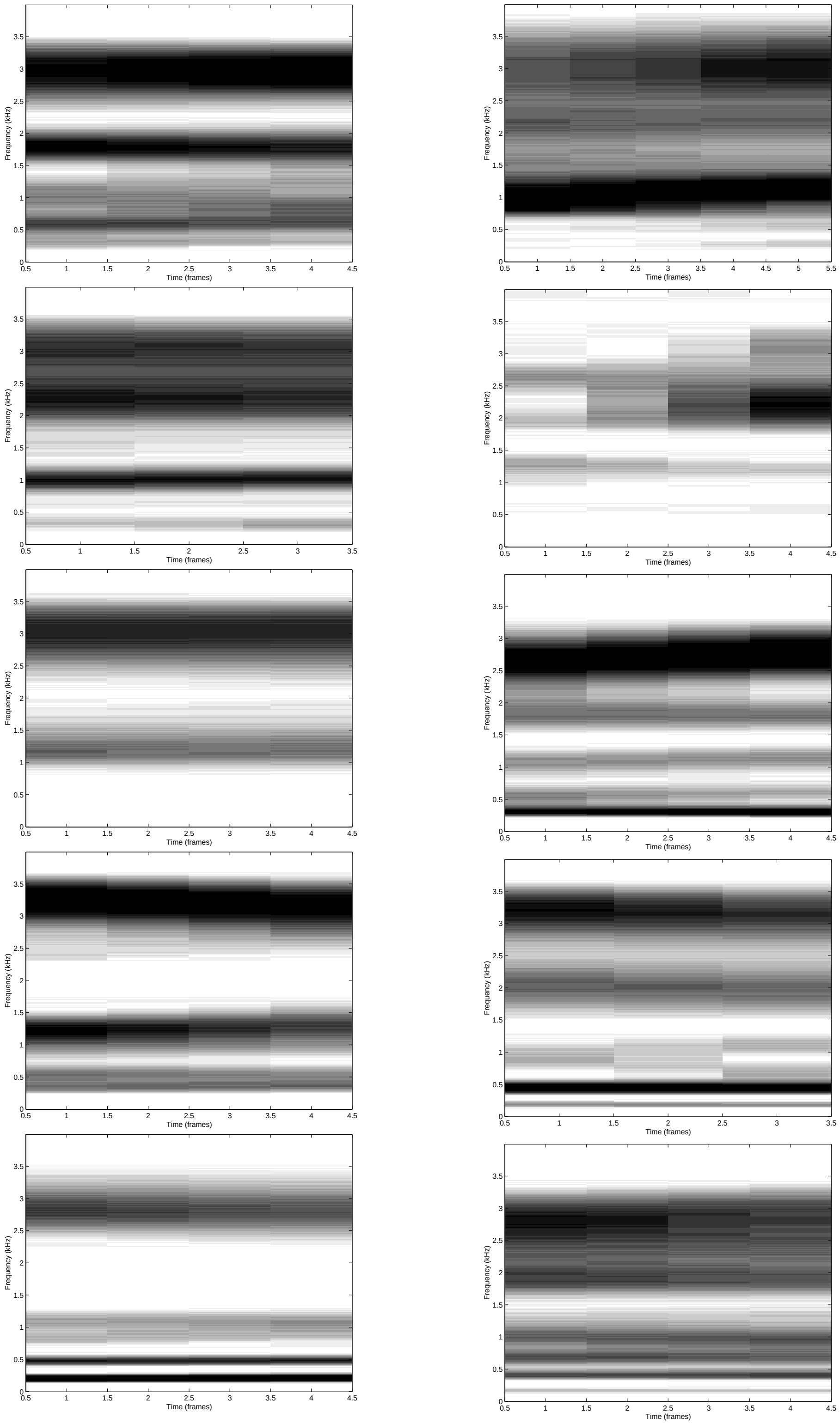


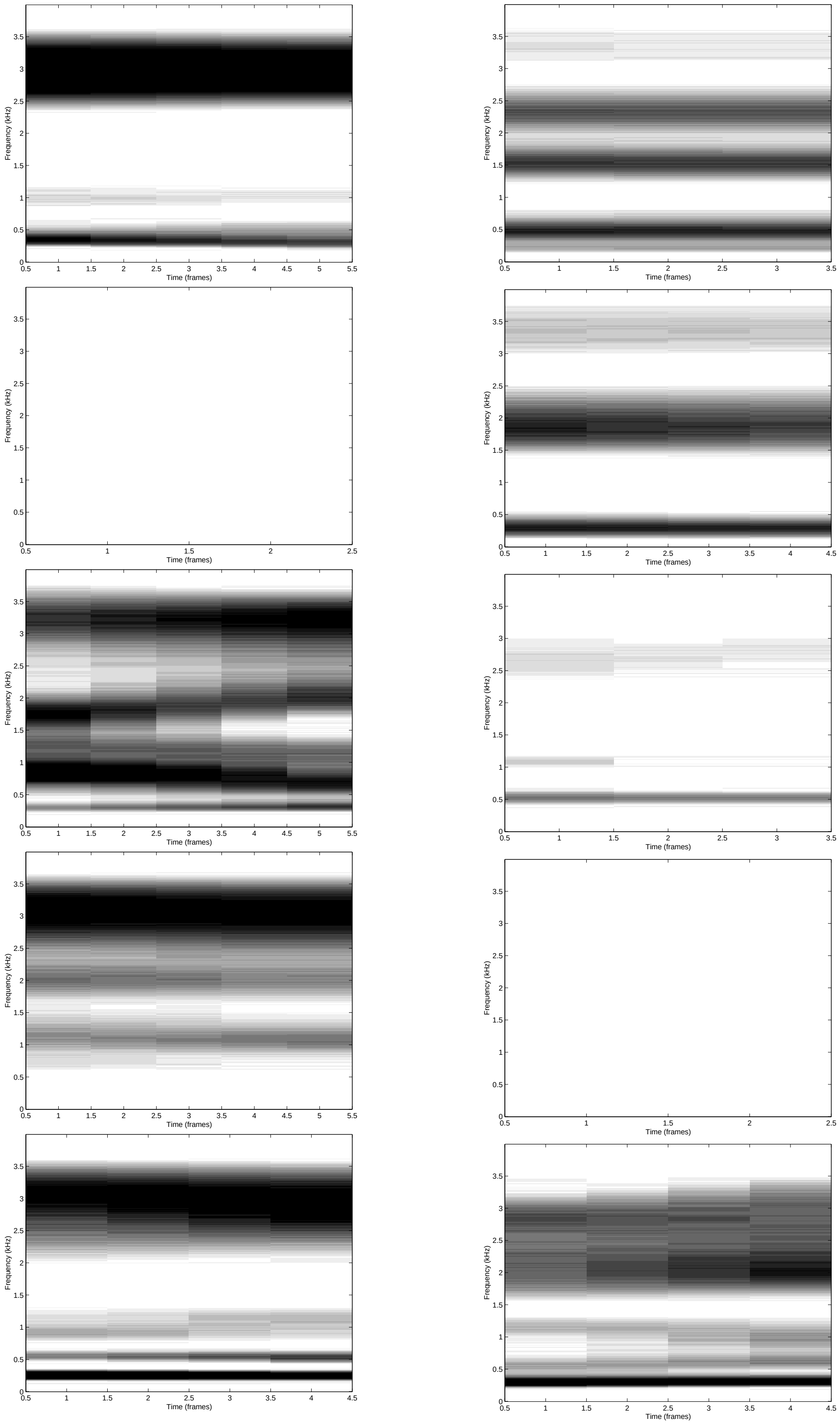
Figure 17: LMScale = 1; LMInsP = 50 .

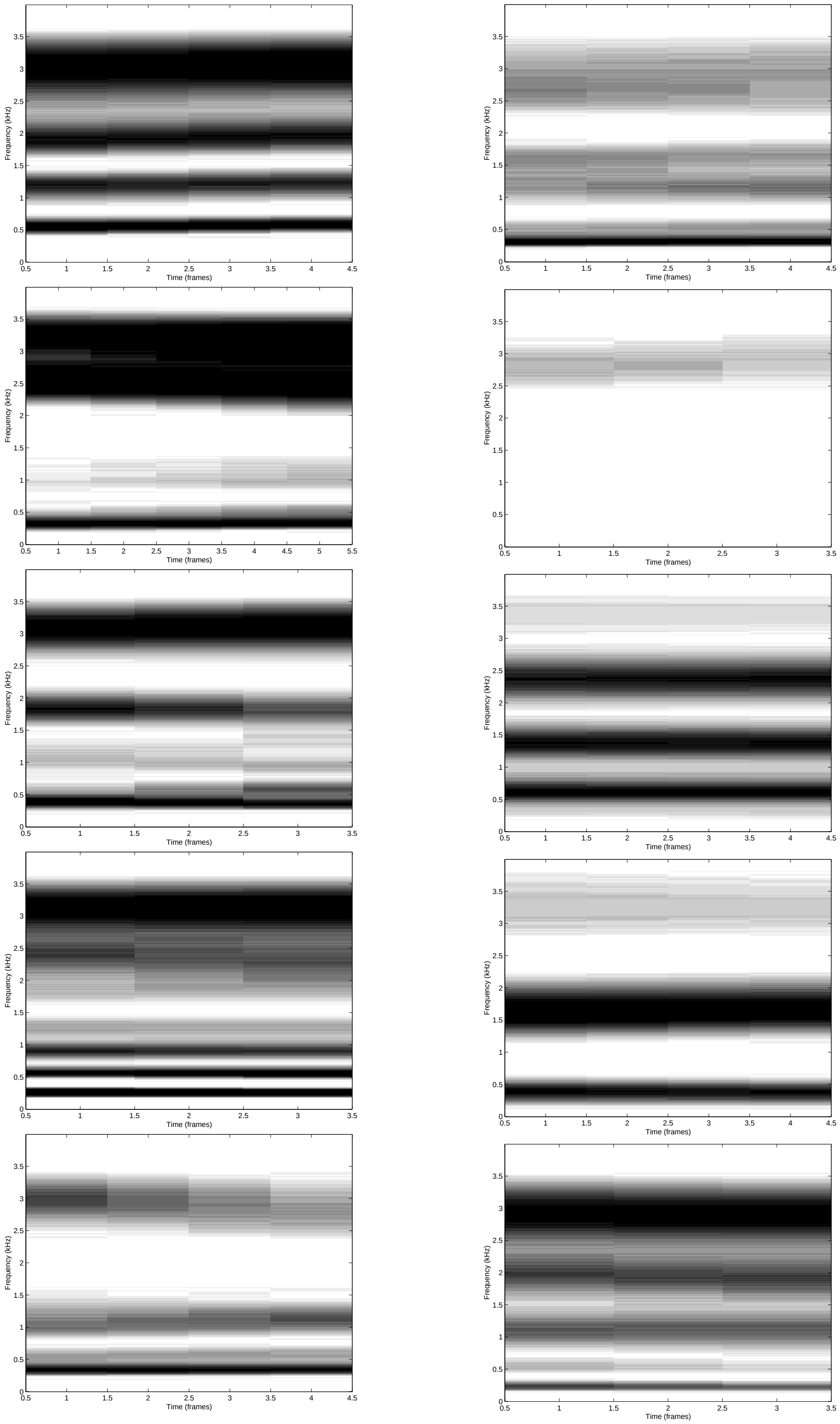
APPENDIX D

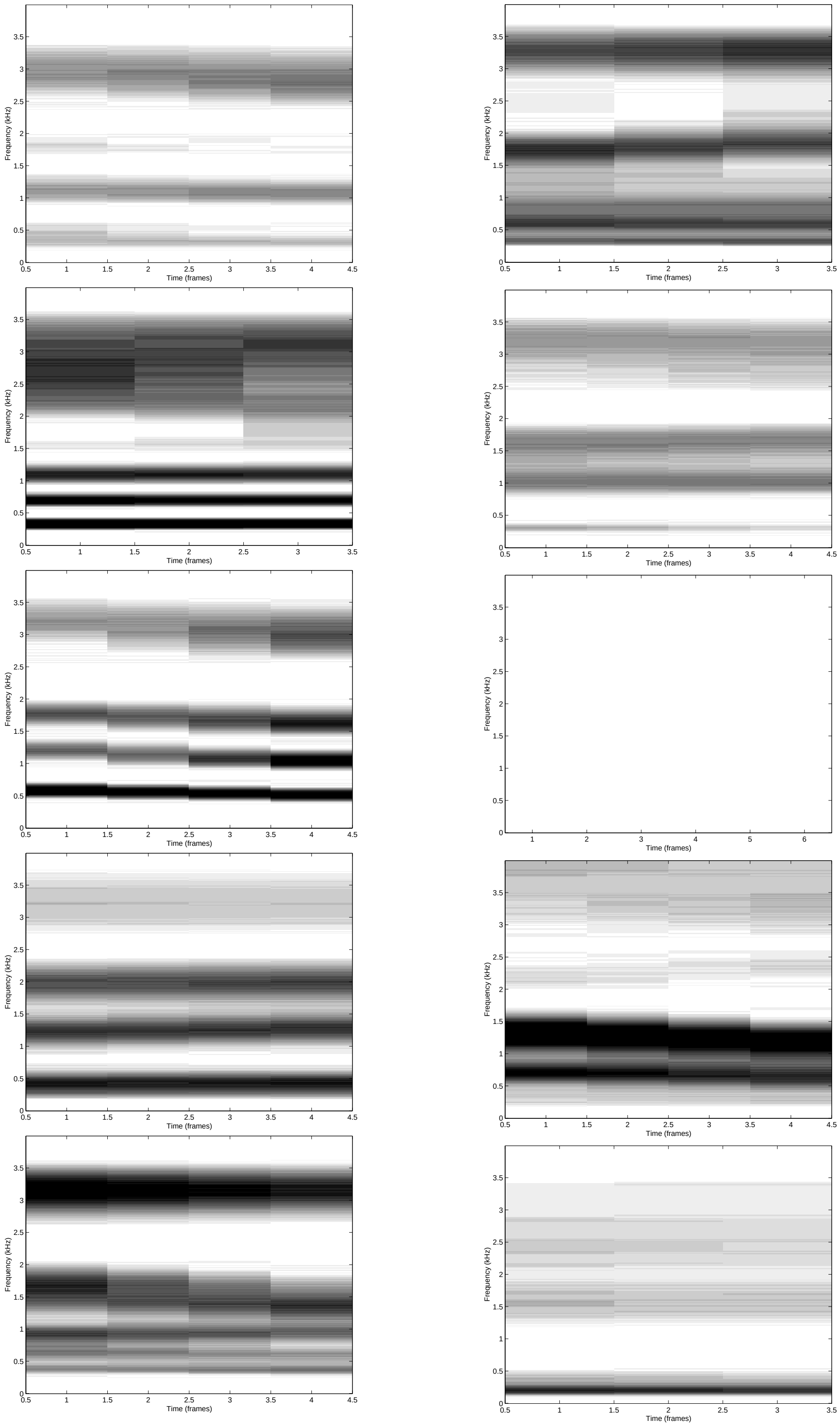
Spectrograms corresponding to trained segments from the SDM for female speaker 5090

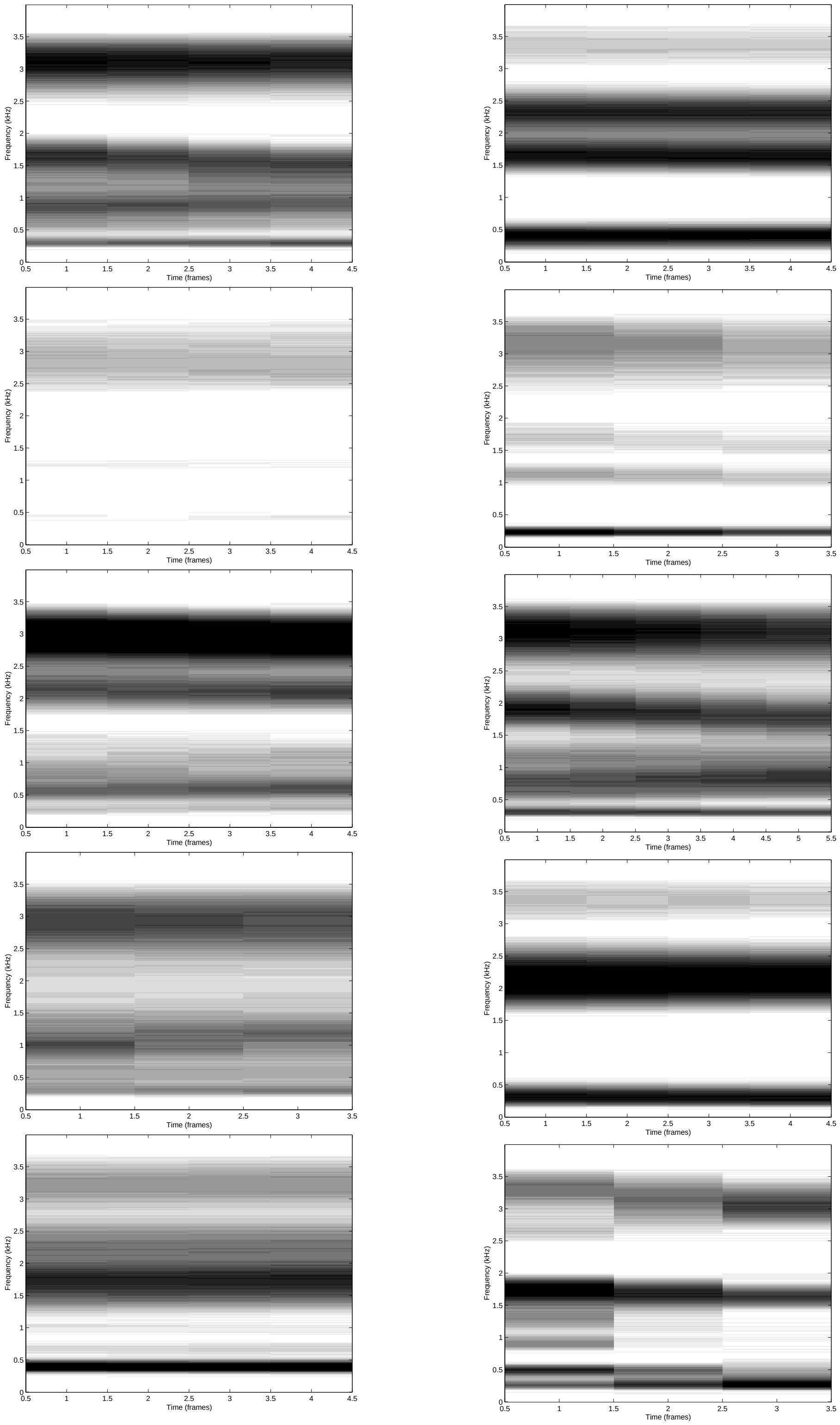

Frequency (kHz)
Time (frames)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

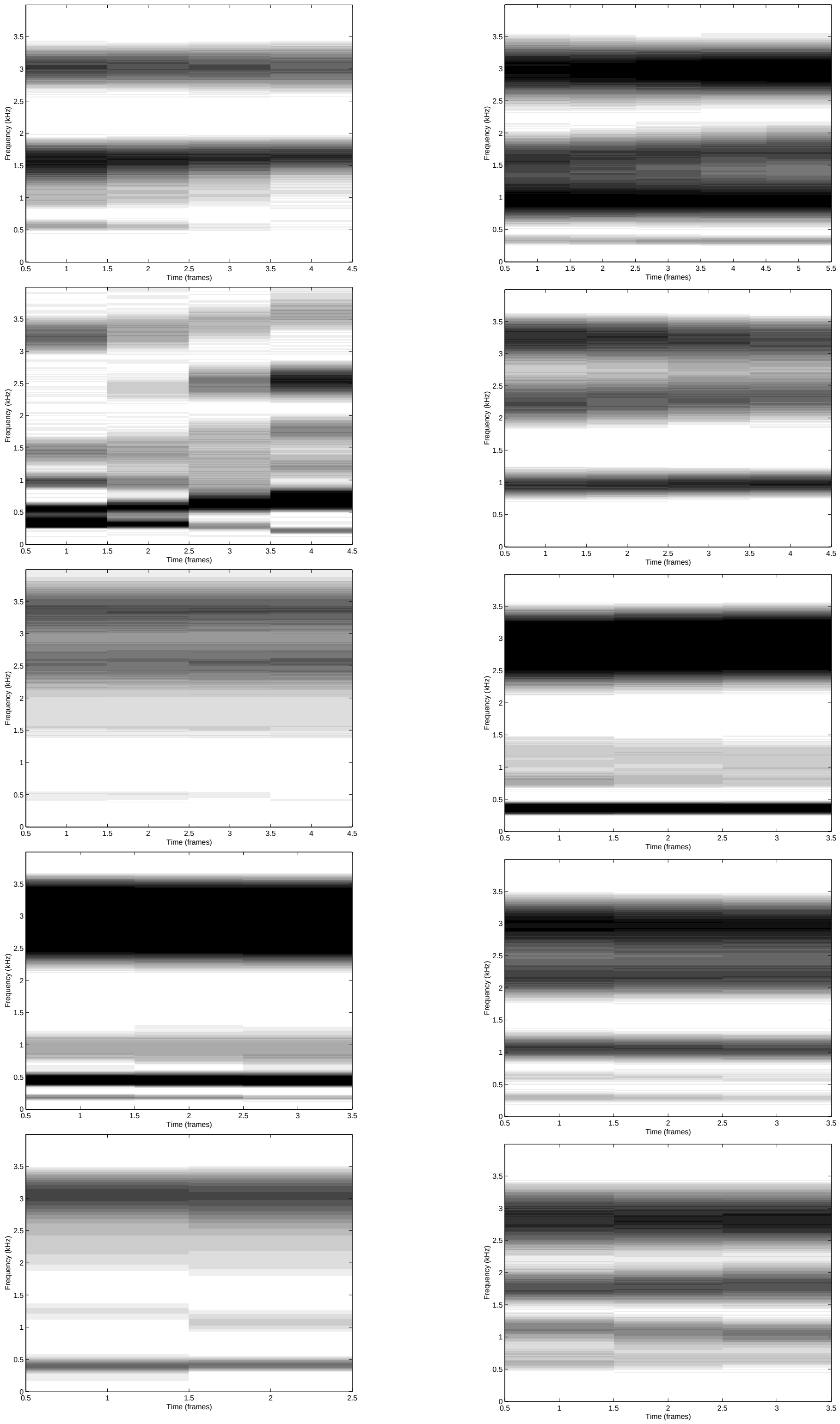

Frequency (kHz)
Time (frames)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5

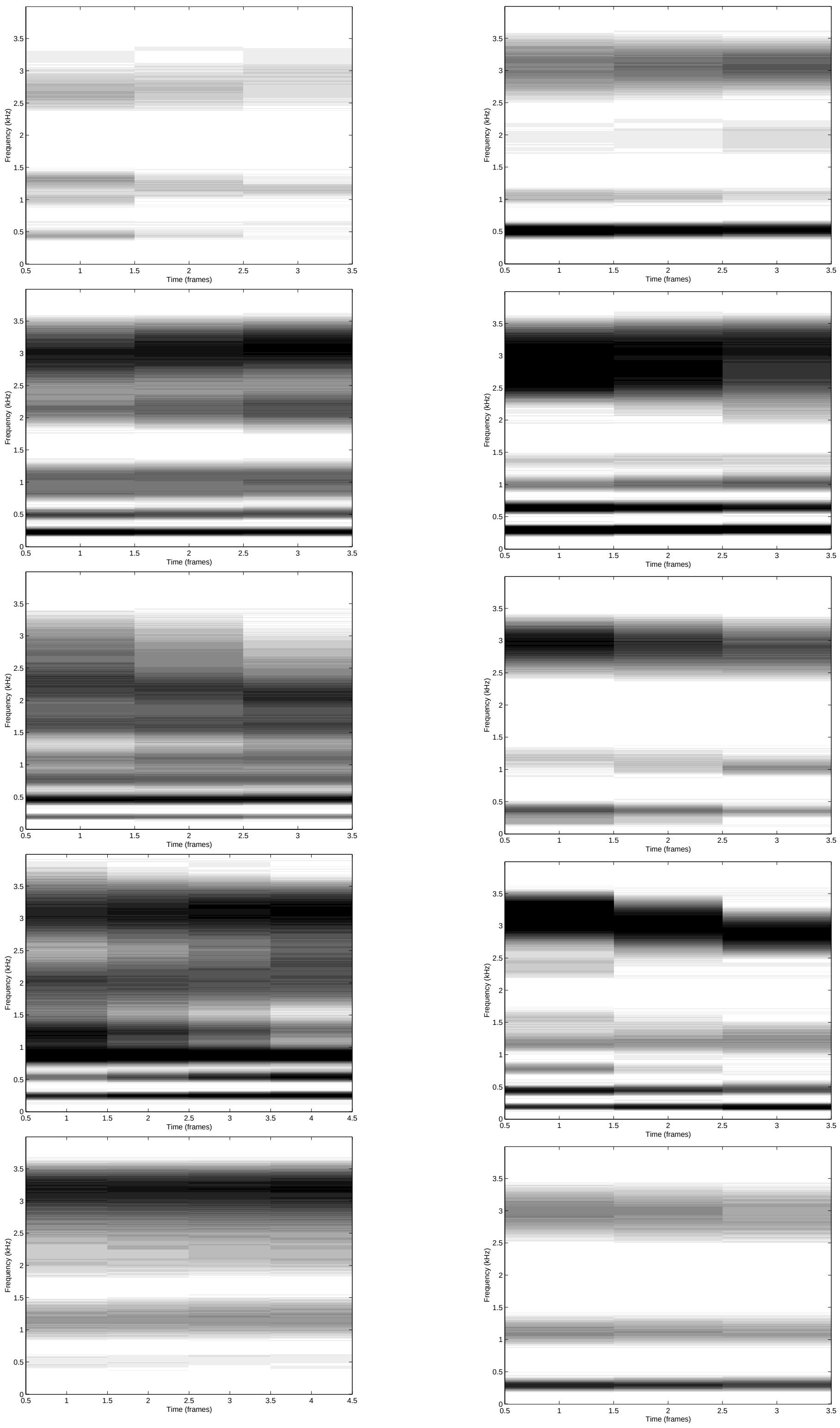

Frequency (kHz)
0
0.5
1
1.5
2
2.5
3
3.5
Time (frames)
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5

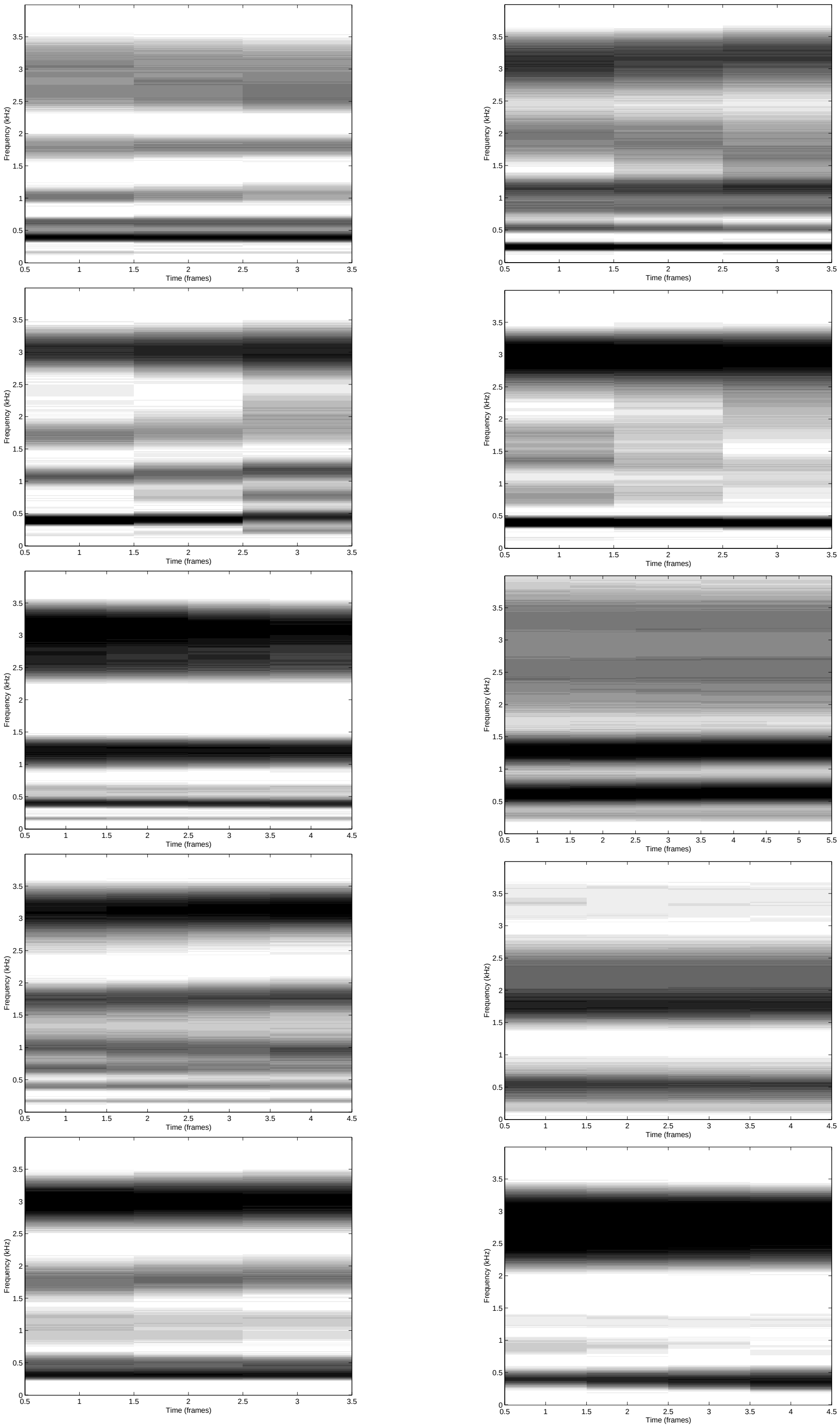

Frequency (kHz)
Time (frames)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5

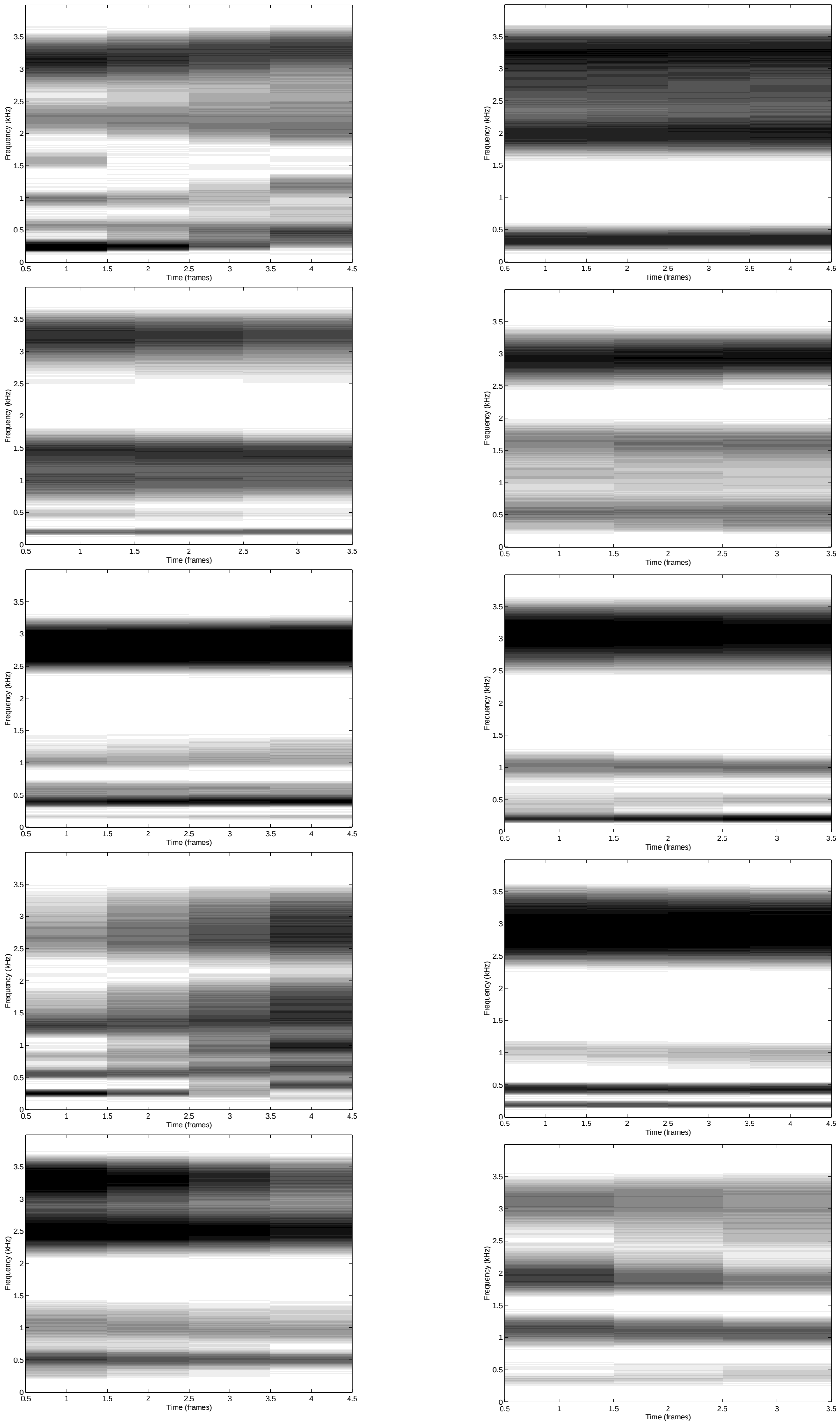

Frequency (kHz)
Time (frames)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
6

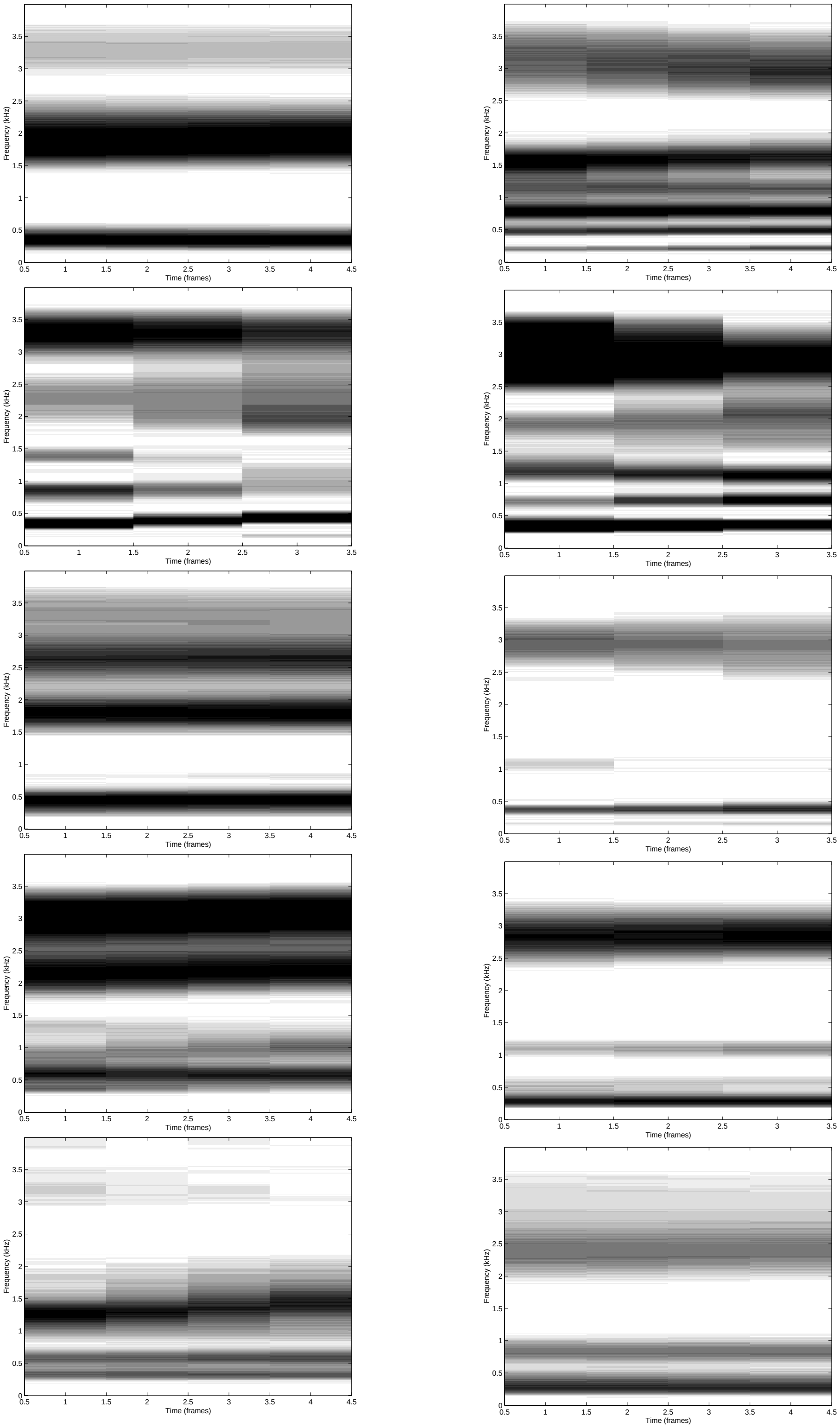

Frequency (kHz)
Time (frames)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5

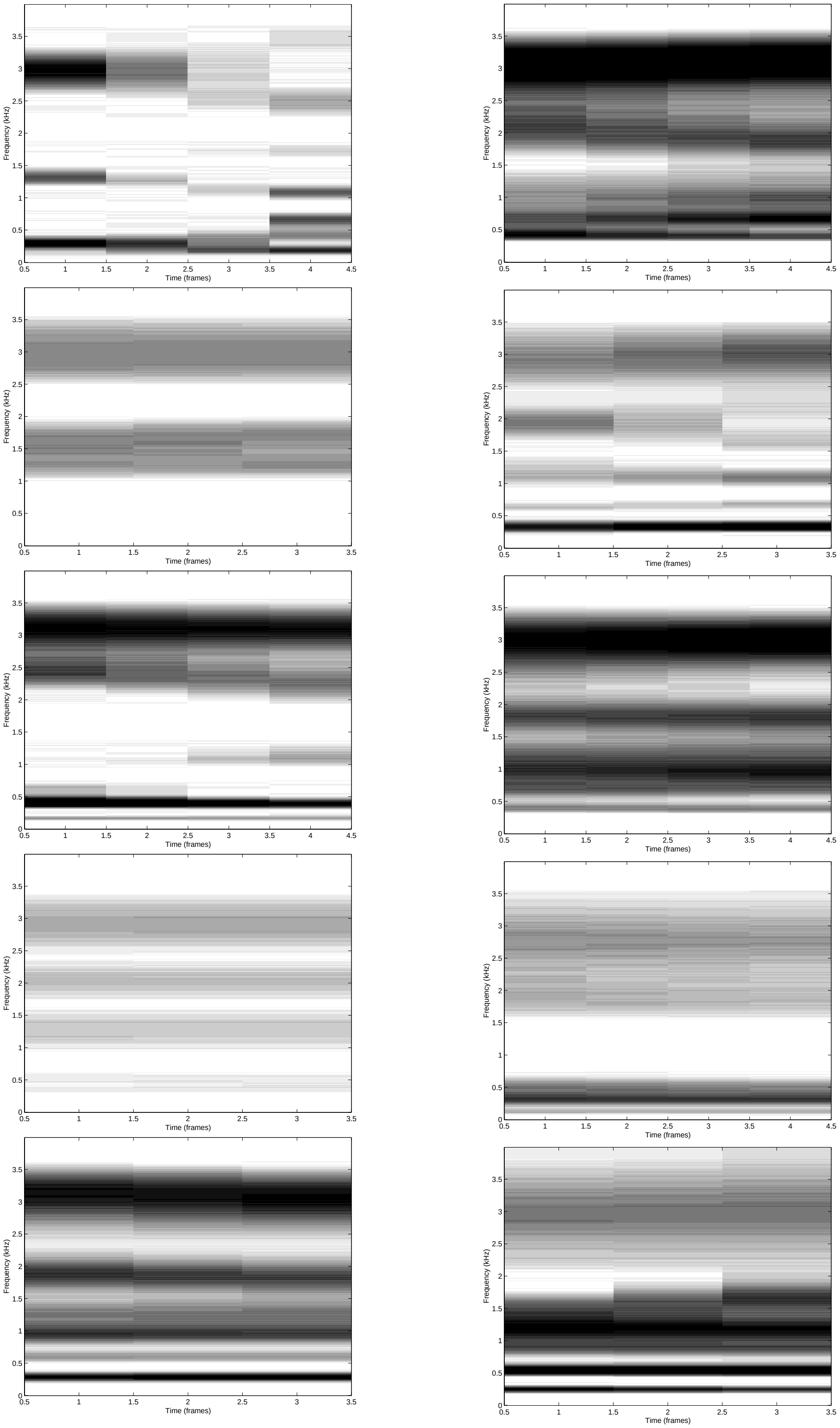

Frequency (kHz)
Time (frames)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5

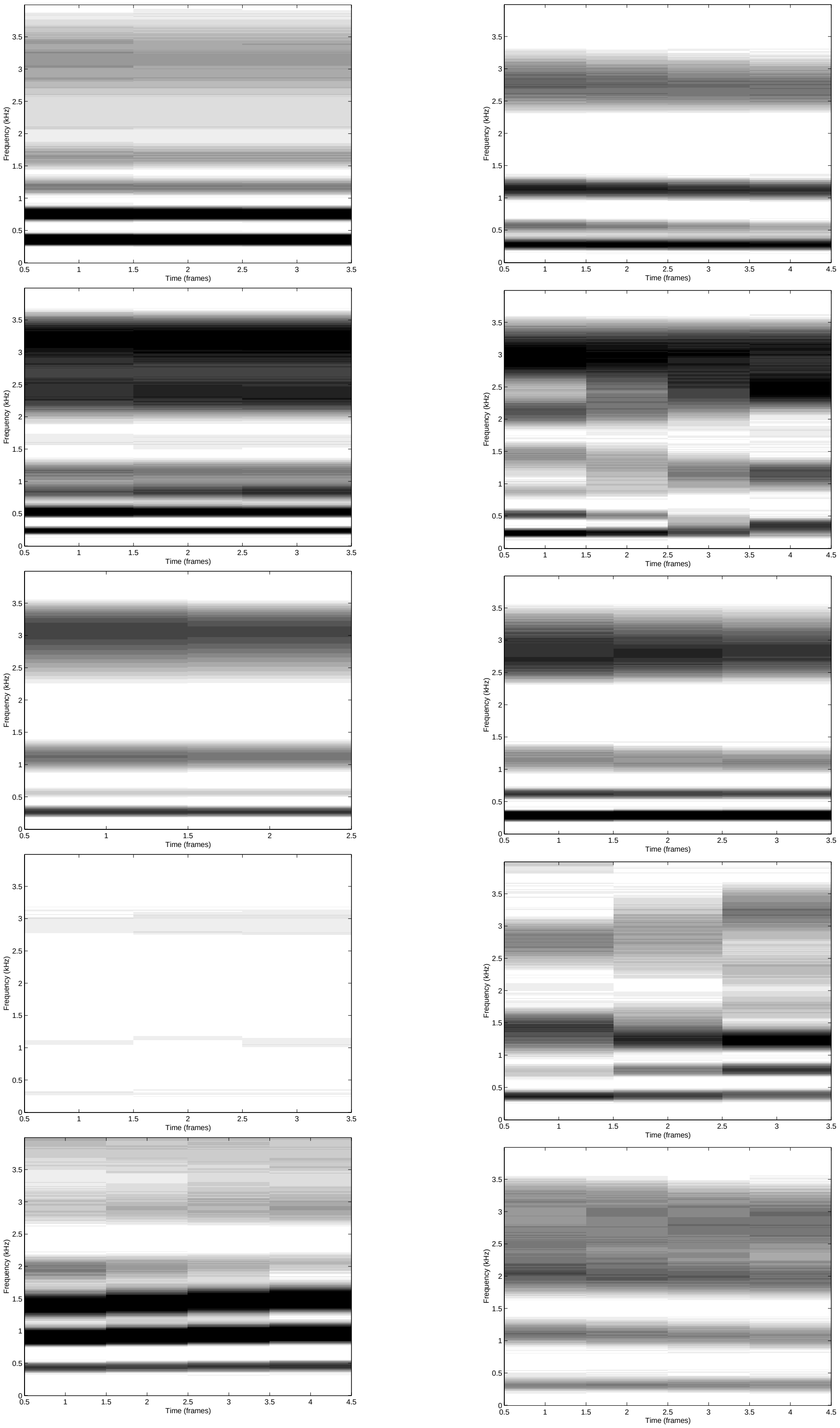

Frequency (kHz)
0
0.5
1
1.5
2
2.5
3
3.5
Time (frames)
0.5
1
1.5
2
2.5
3
3.5
4
4.5

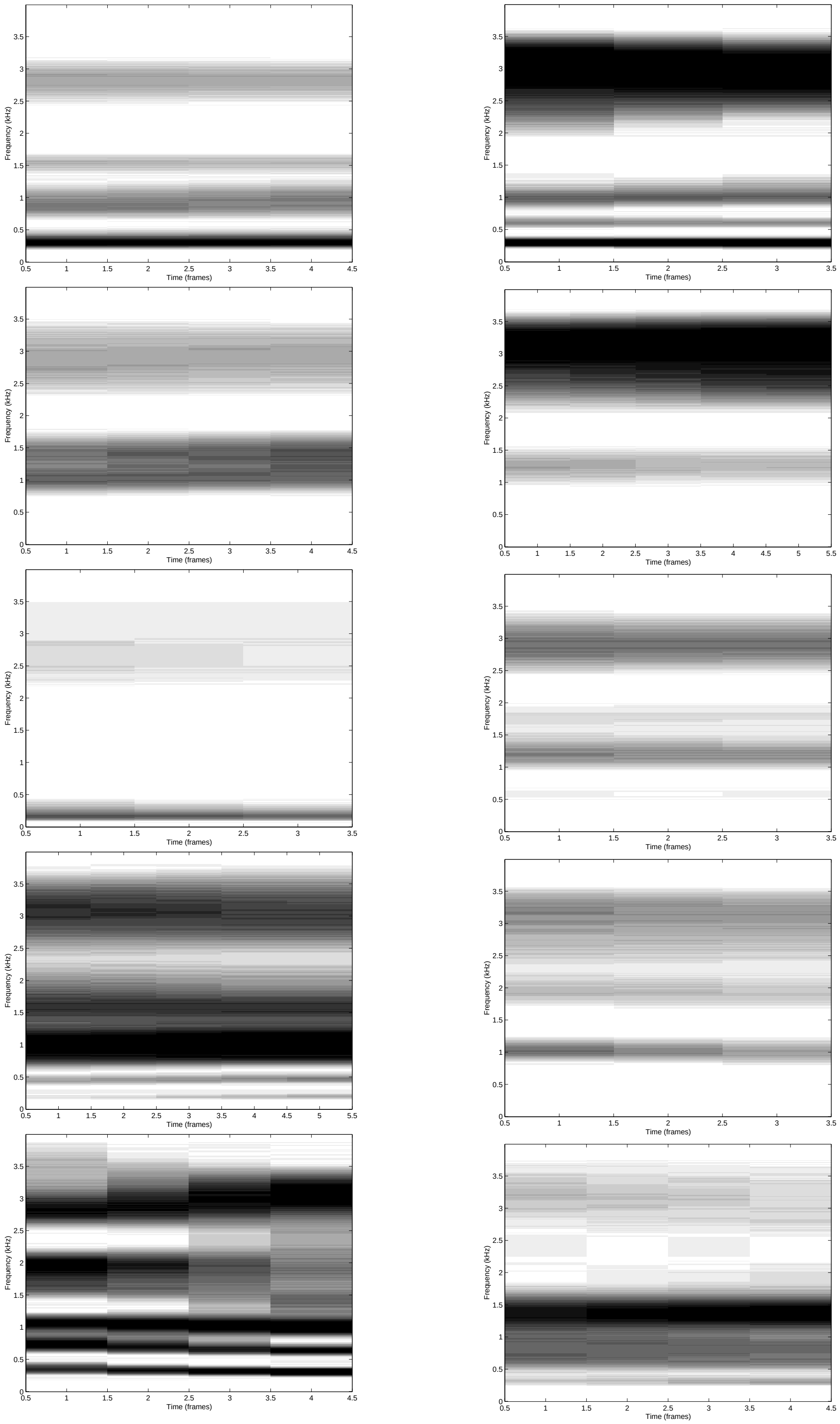

Frequency (kHz)
0
0.5
1
1.5
2
2.5
3
3.5
Time (frames)
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5

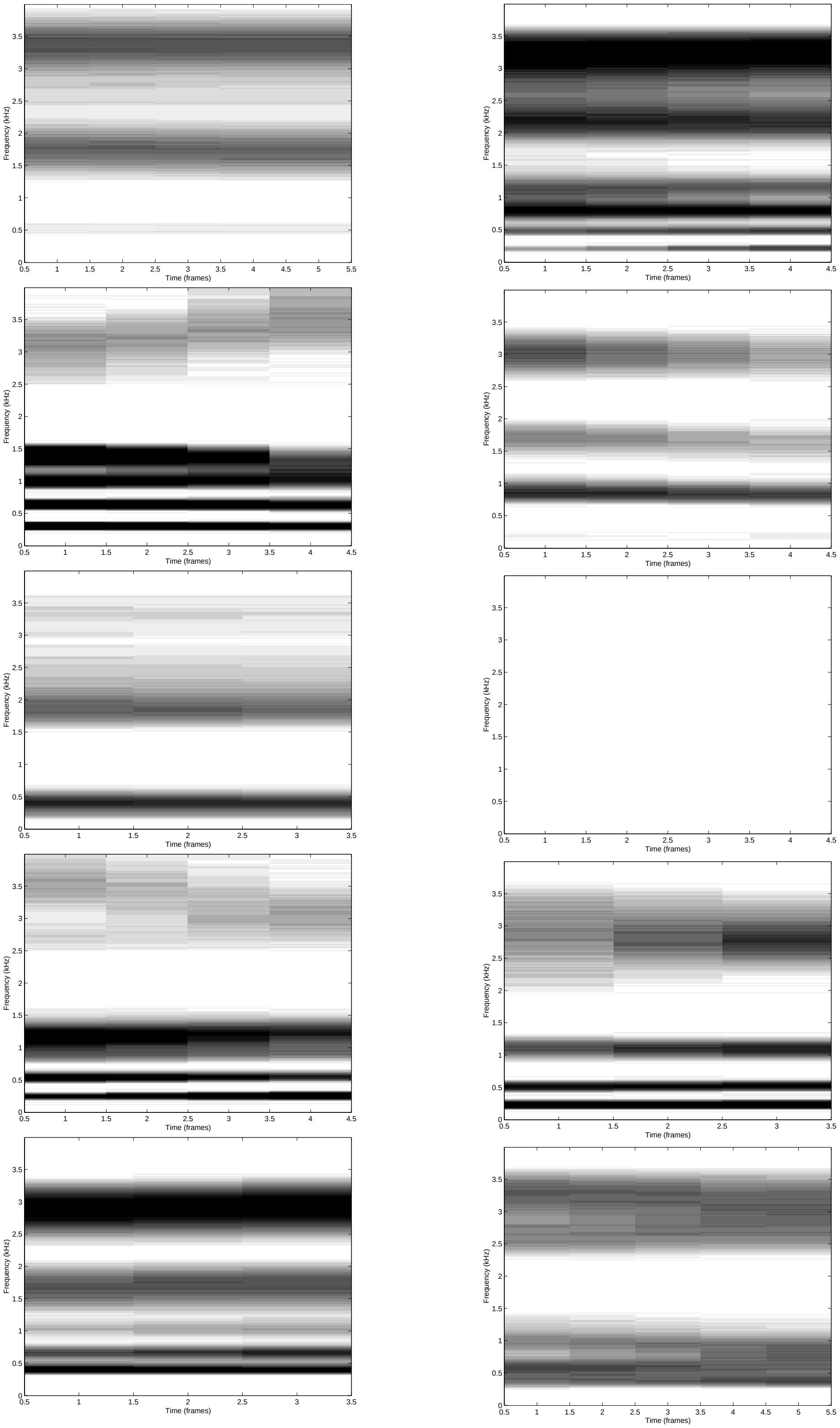

Frequency (kHz)
0
0.5
1
1.5
2
2.5
3
3.5
Time (frames)
0.5
1
1.5
2
2.5
3
3.5
4
4.5

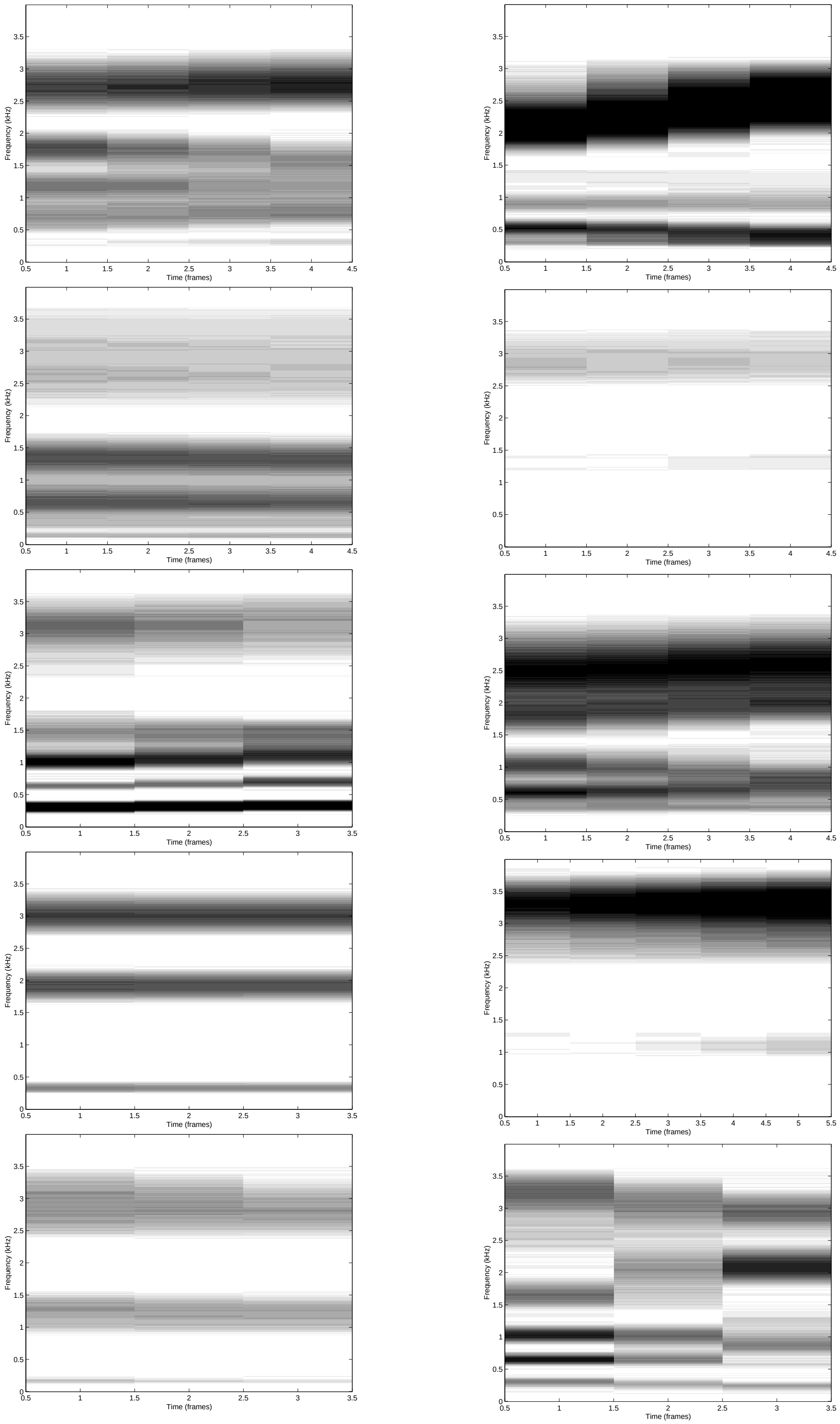

Frequency (kHz)
0
0.5
1
1.5
2
2.5
3
3.5
Time (frames)
0.5
1
1.5
2
2.5
3
3.5
4
4.5

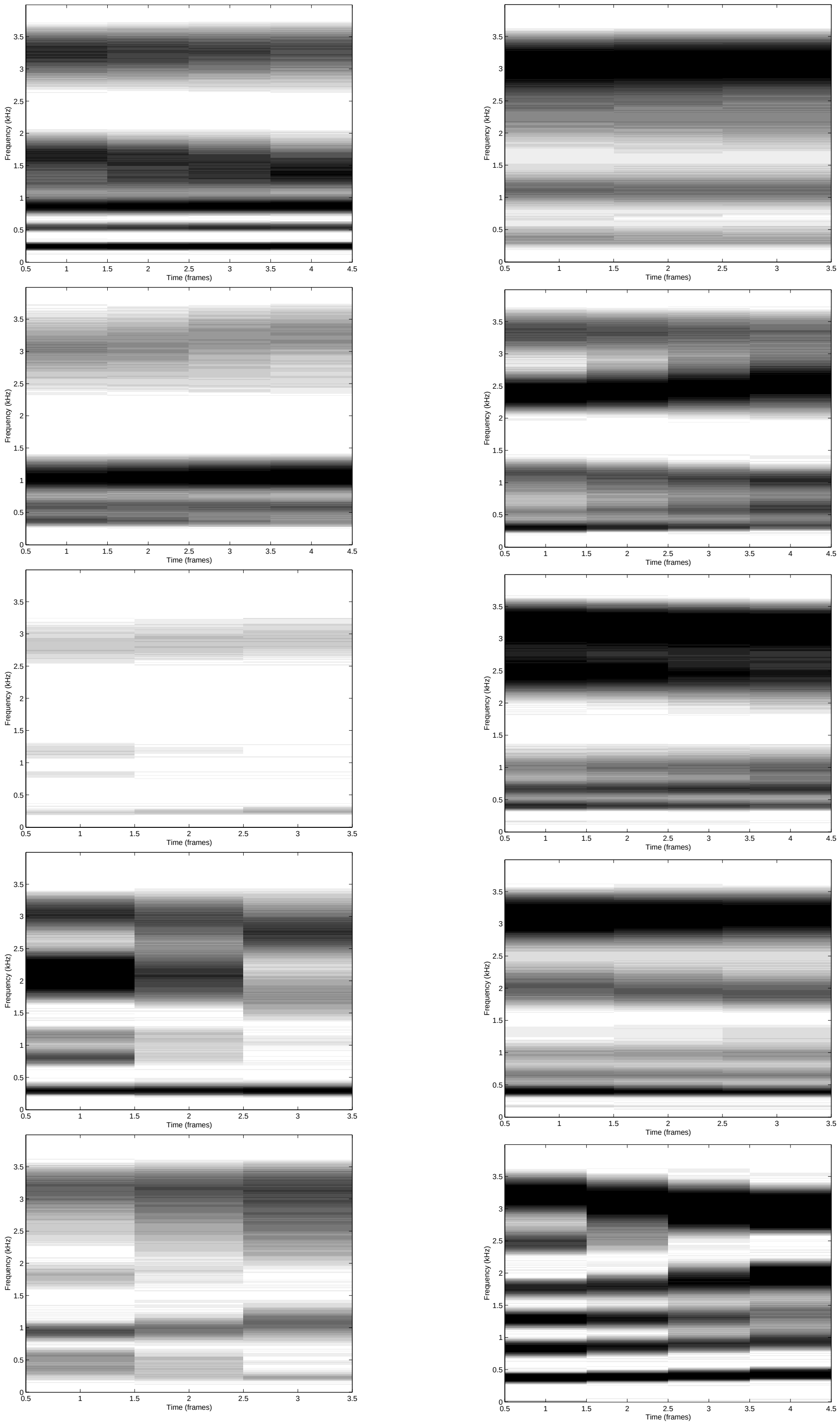

Frequency (kHz)
Time (frames)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

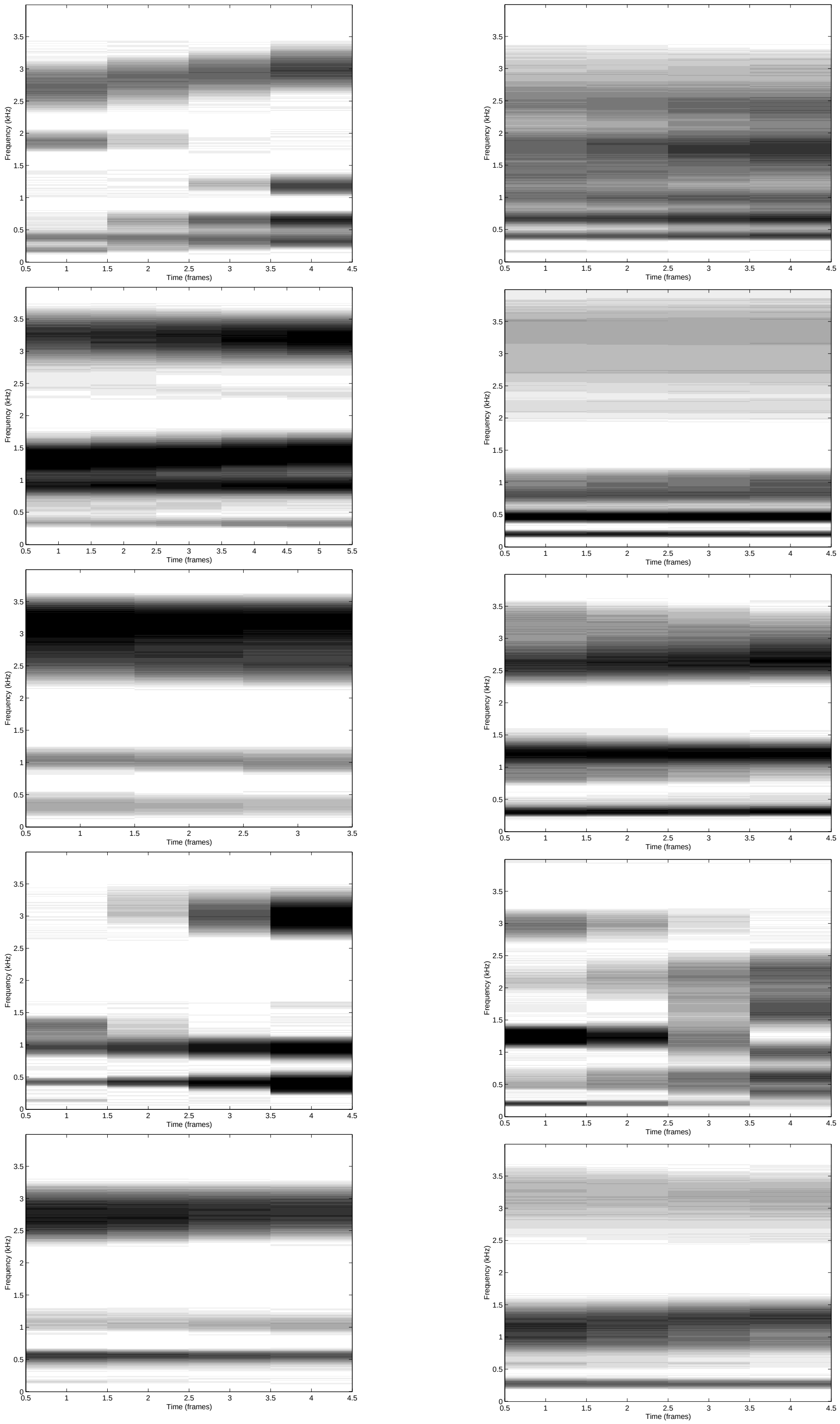

Frequency (kHz)
Time (frames)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

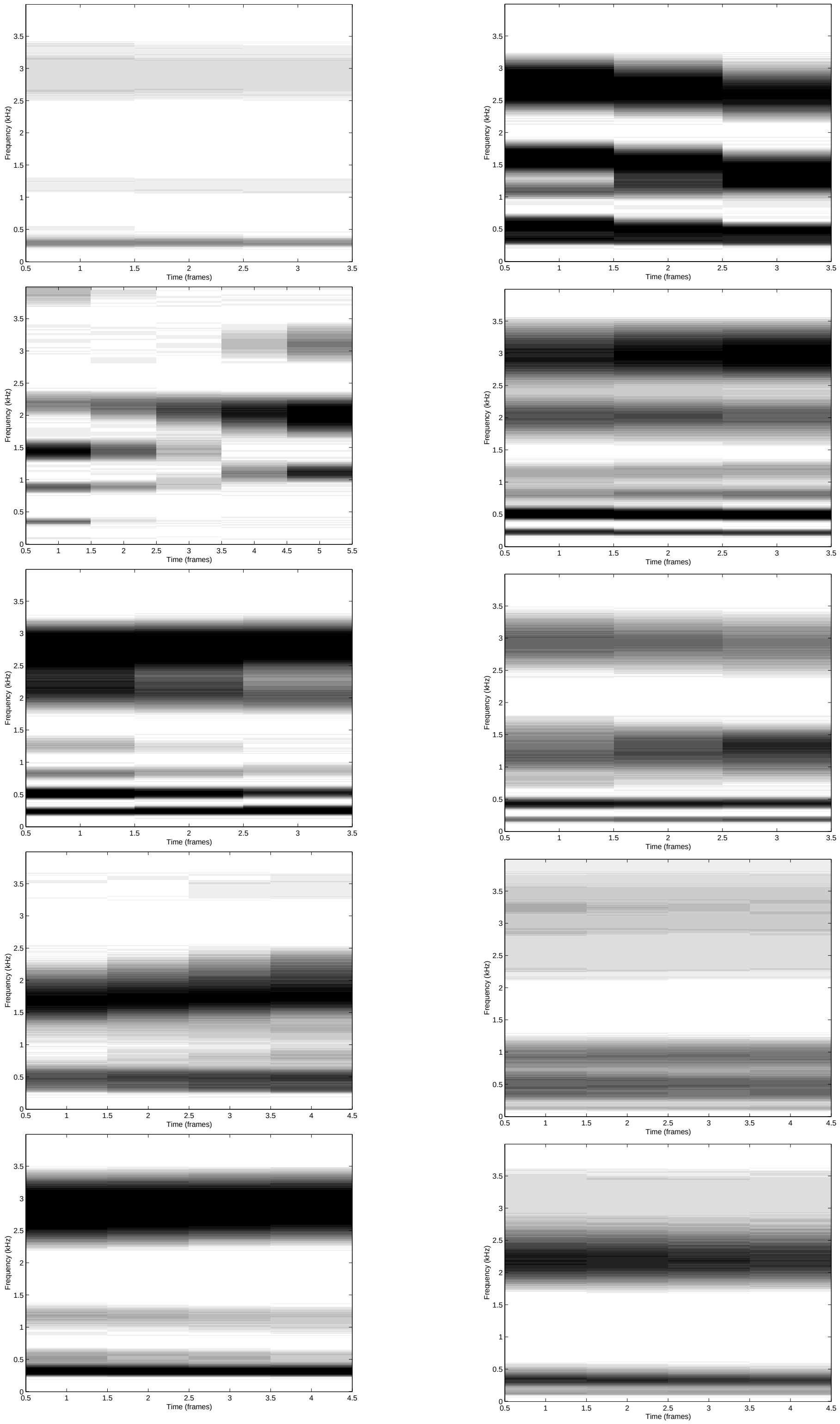

Frequency (kHz)
0
0.5
1
1.5
2
2.5
3
3.5
Time (frames)
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5

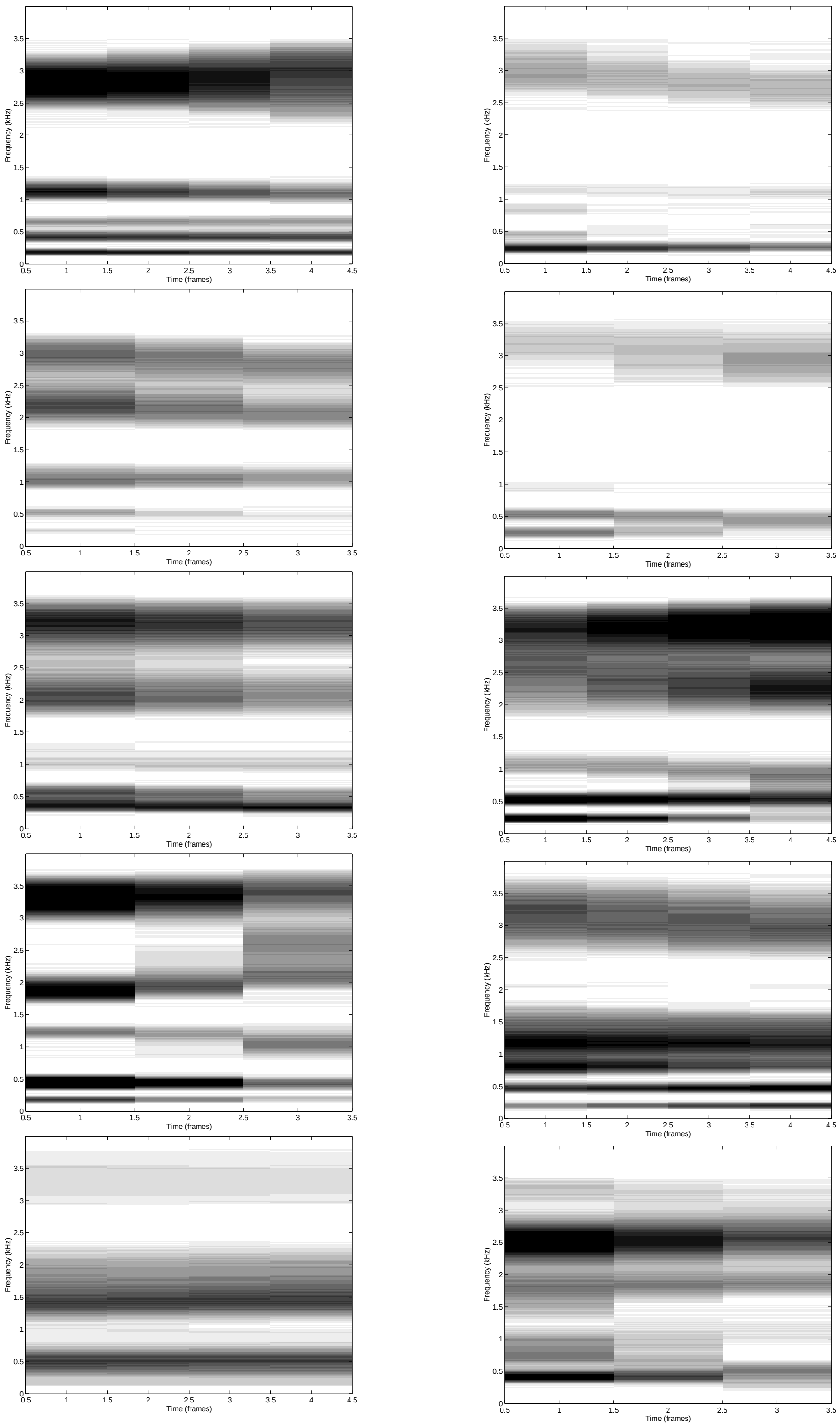

Frequency (kHz)
Time (frames)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5

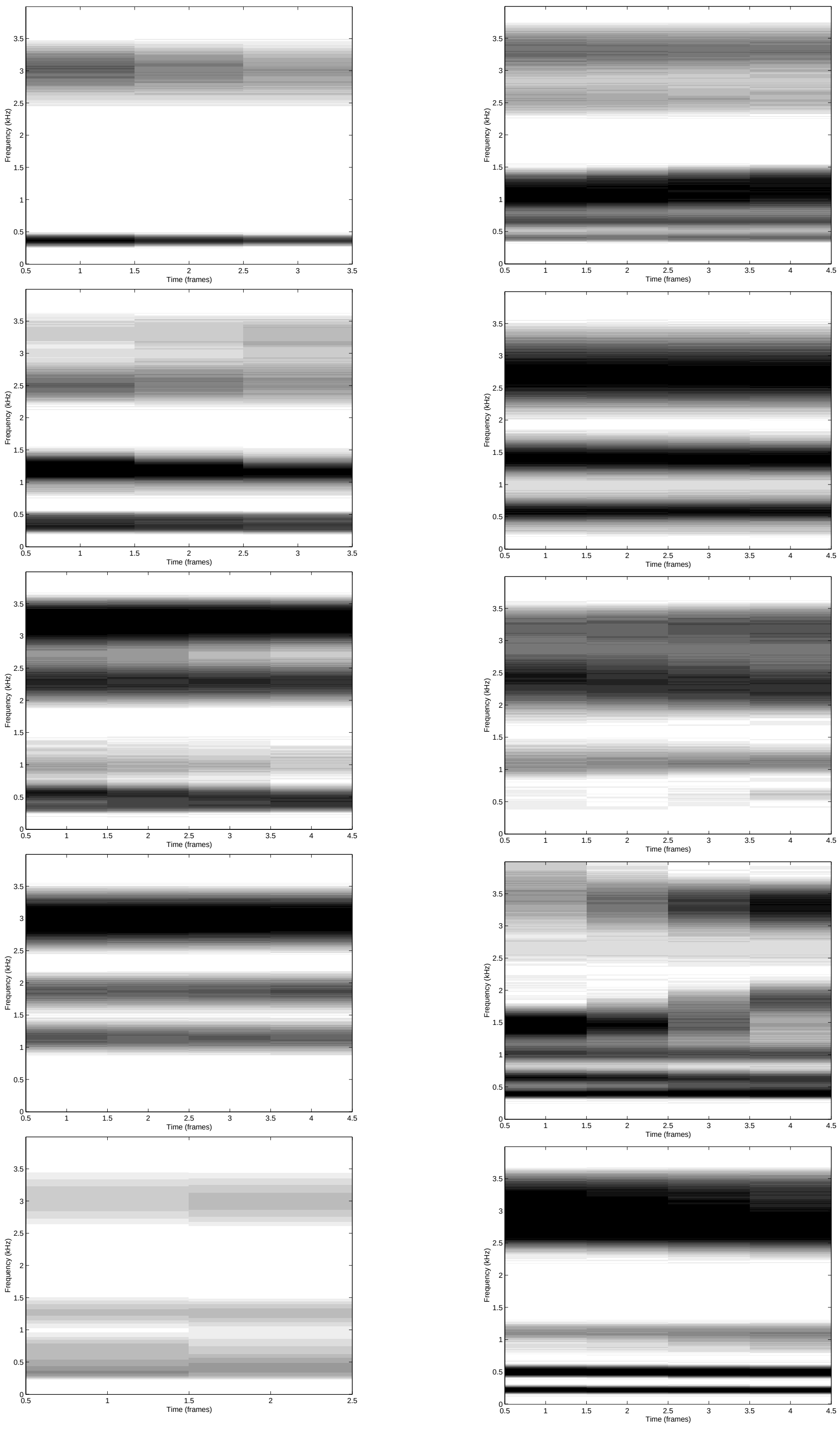

Frequency (kHz)
Time (frames)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5

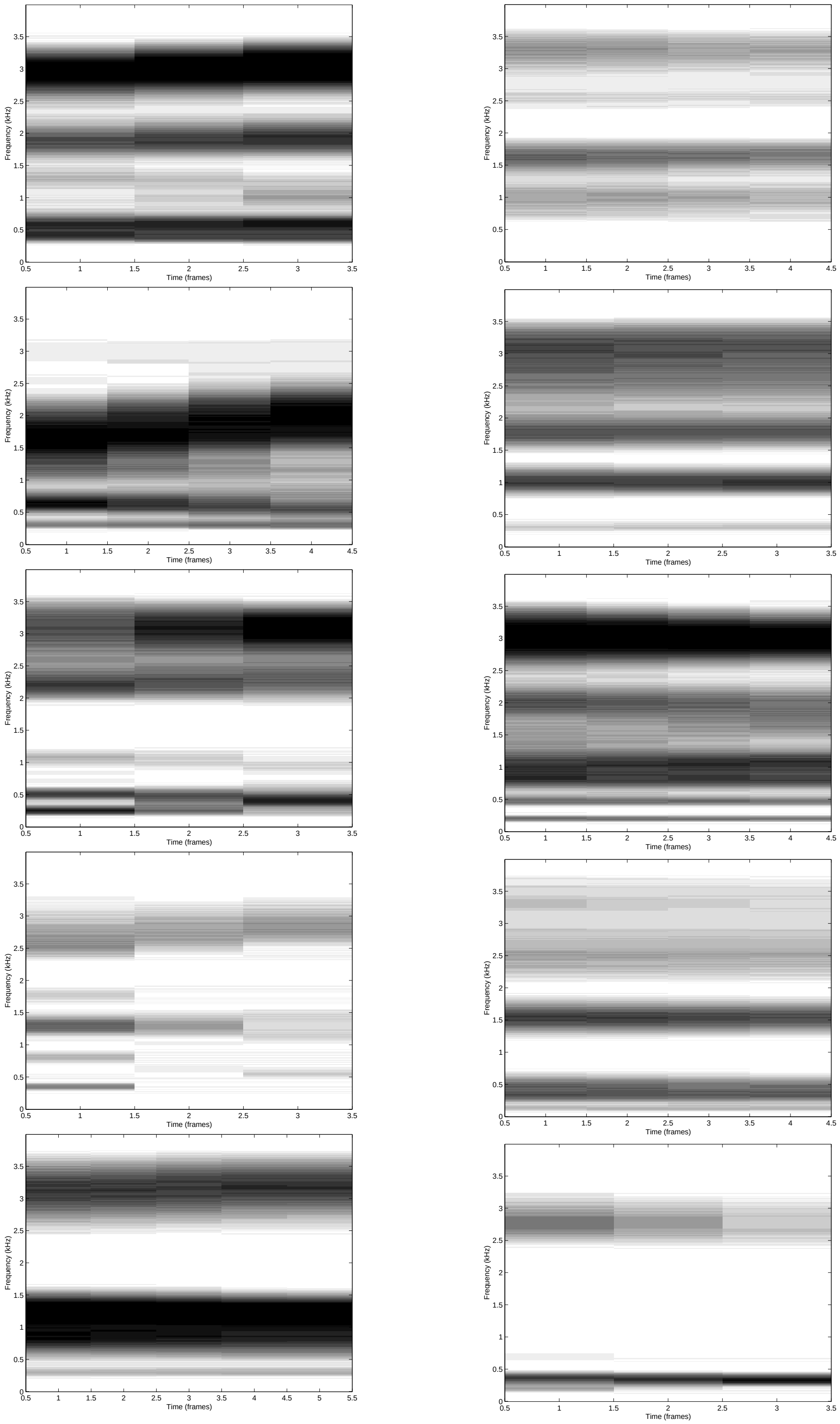

Frequency (kHz)
Time (frames)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

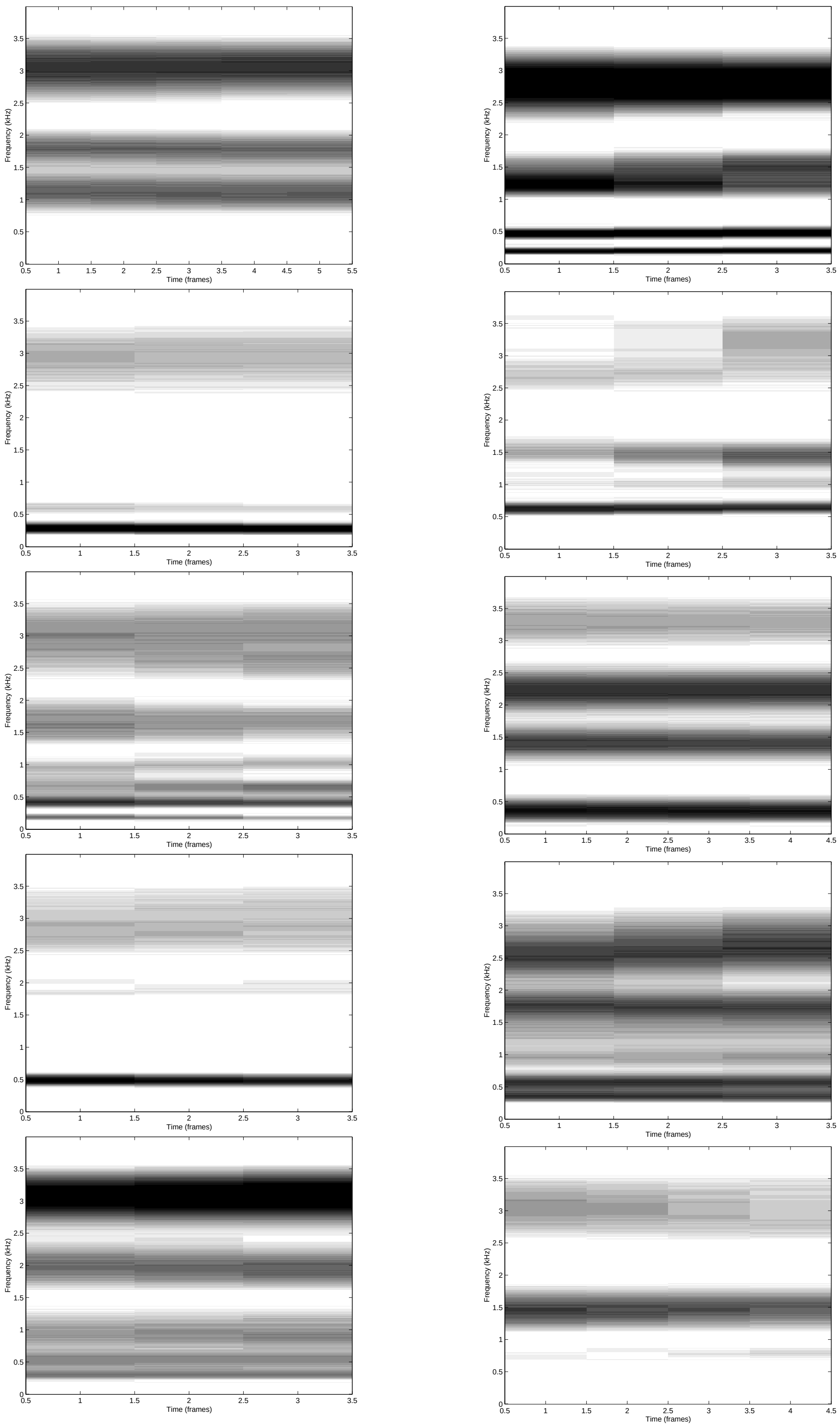

Frequency (kHz)
Time (frames)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5

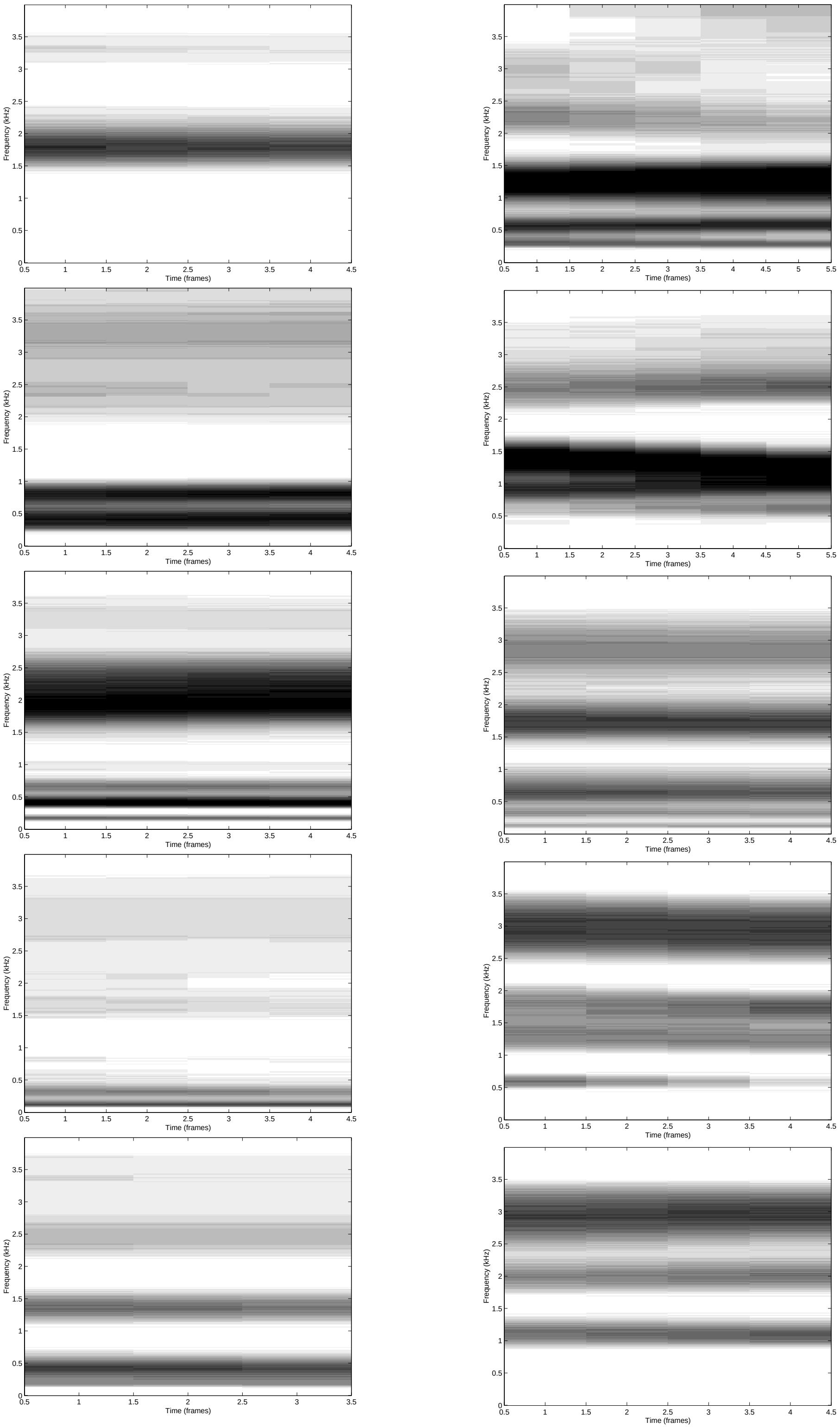

Frequency (kHz)
Time (frames)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5

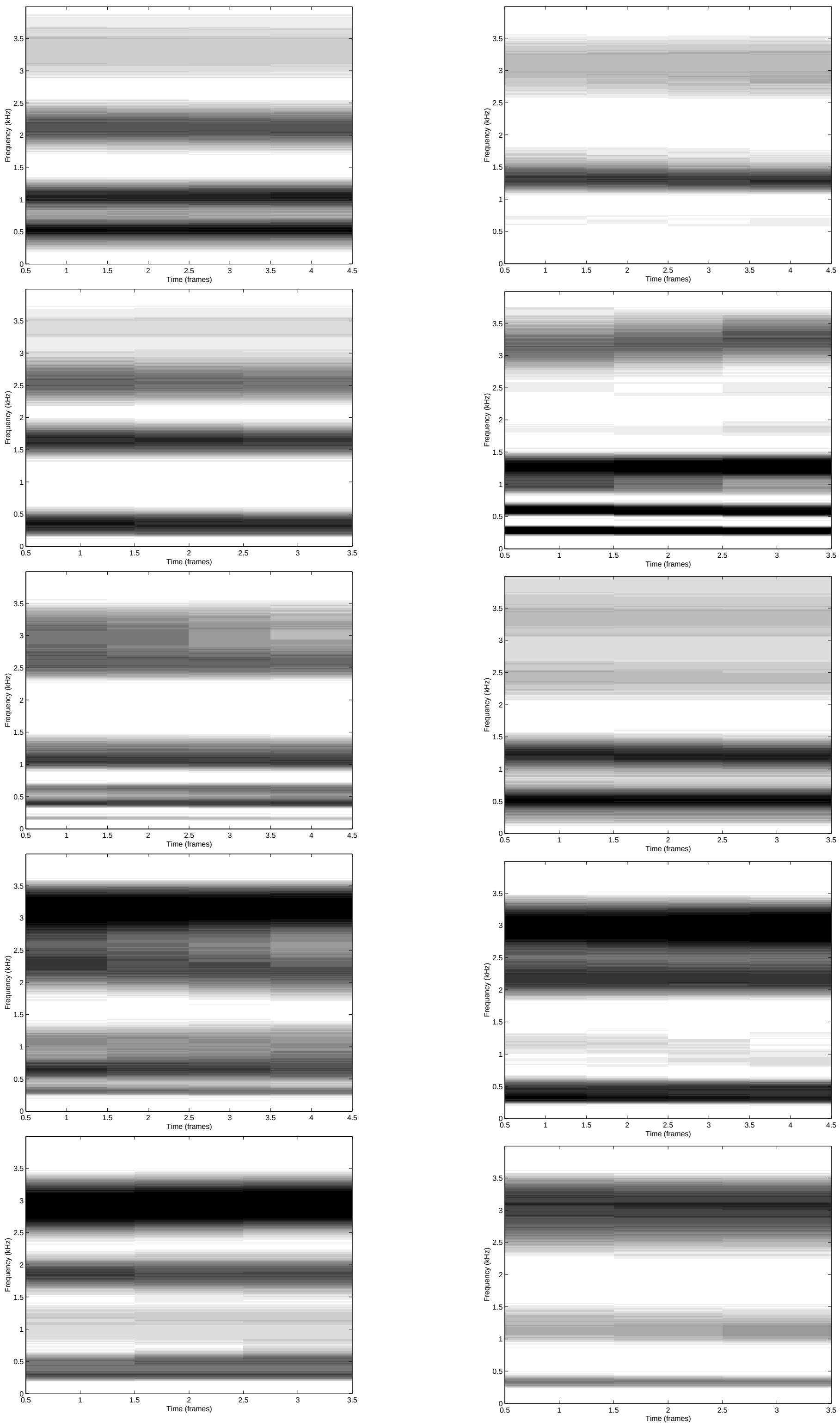

Frequency (kHz)
Time (frames)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

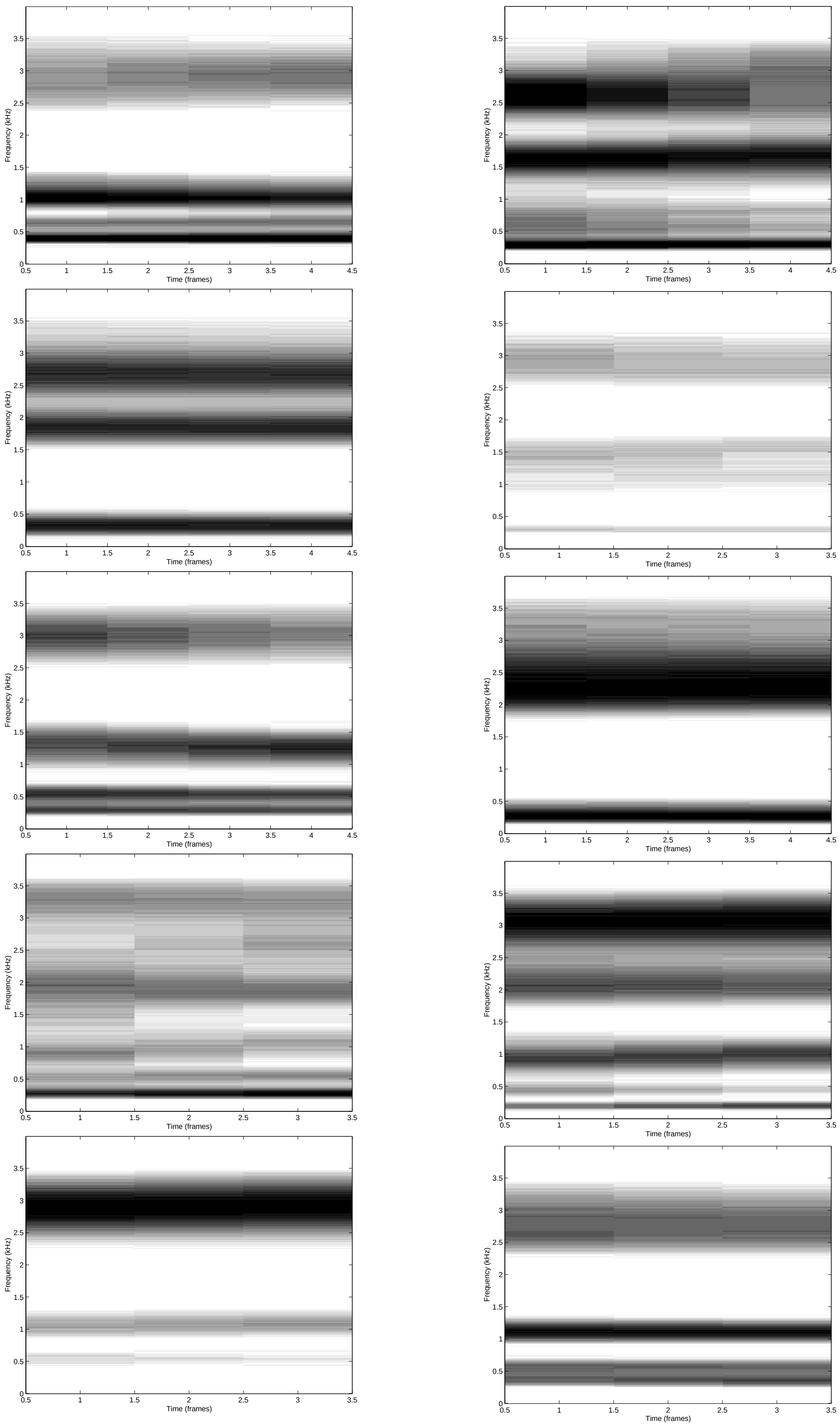

Frequency (kHz)
Time (frames)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

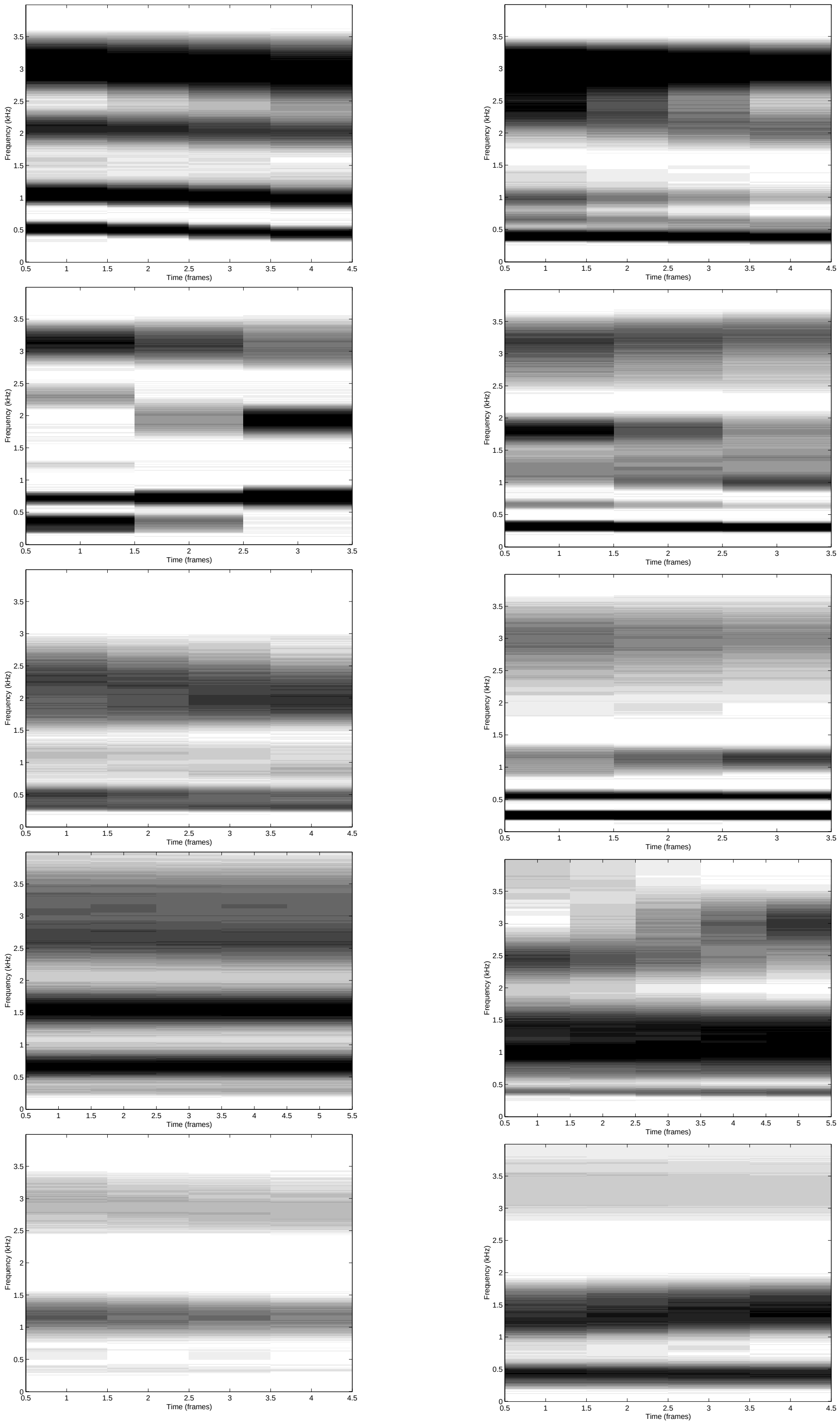

Frequency (kHz)
Time (frames)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5

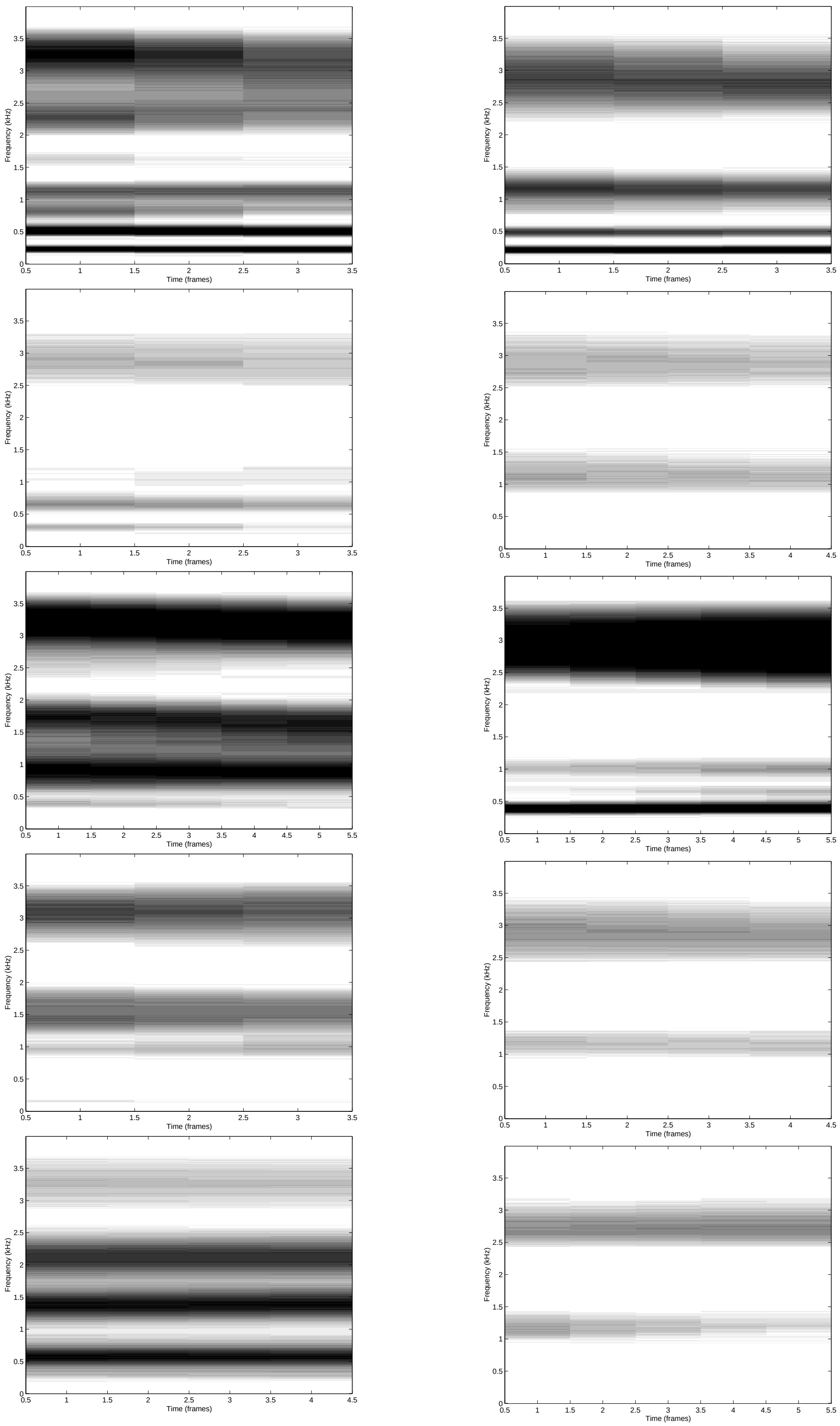

Frequency (kHz)
0
0.5
1
1.5
2
2.5
3
3.5
Time (frames)
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5


Frequency (kHz)
0
0.5
1
1.5
2
2.5
3
3.5
Time (frames)
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5

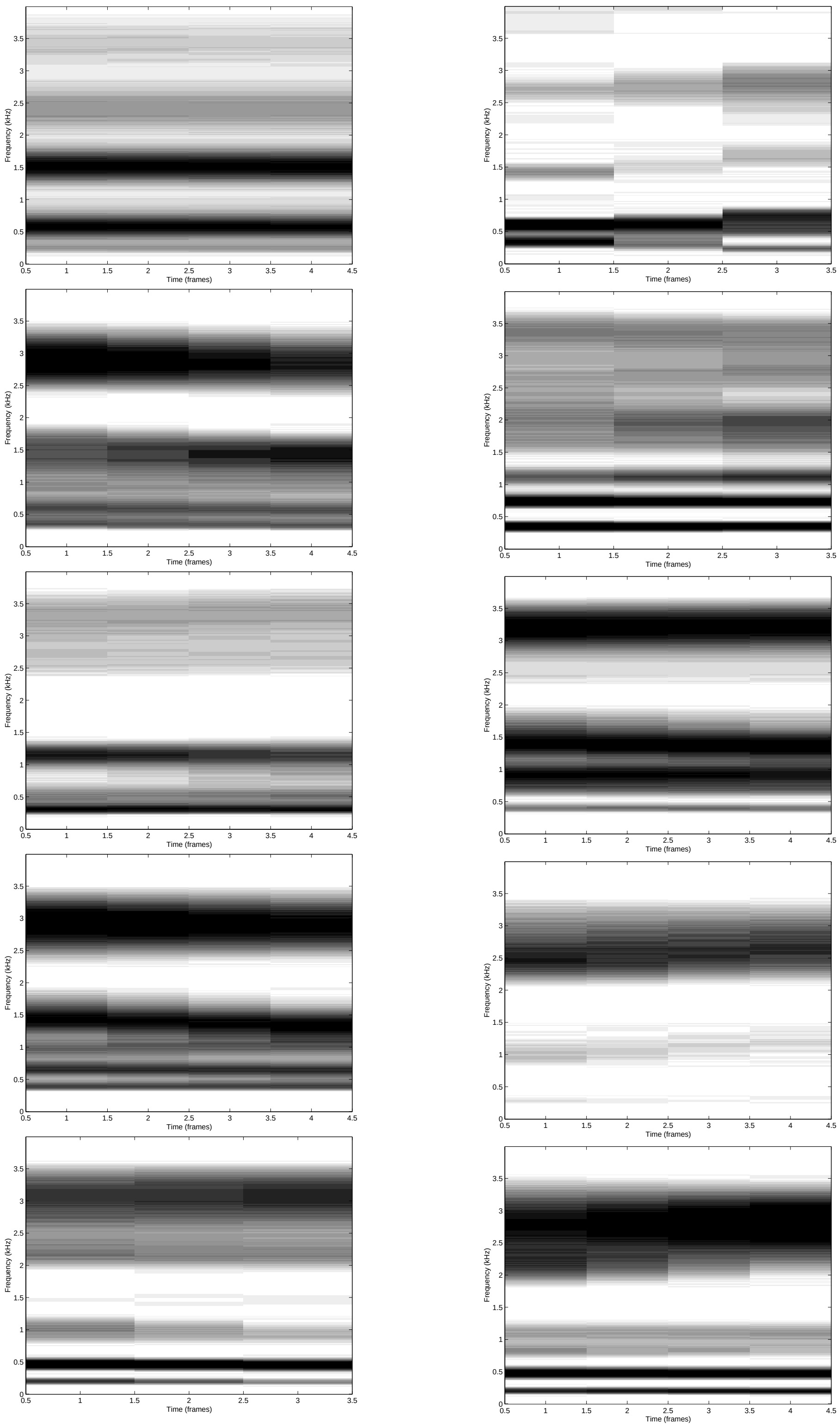

Frequency (kHz)
0
0.5
1
1.5
2
2.5
3
3.5
Time (frames)
0.5
1
1.5
2
2.5
3
3.5
4
4.5

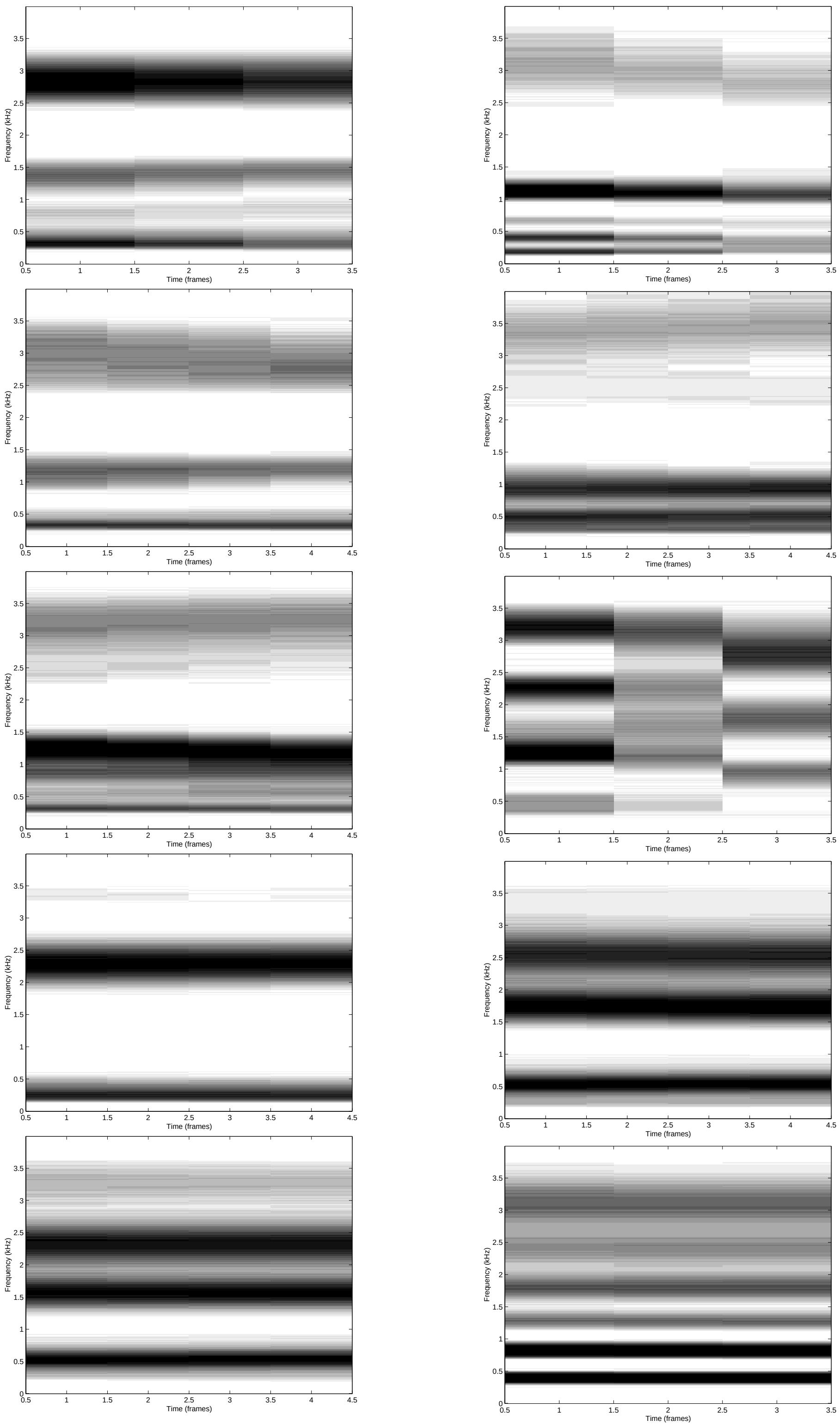

Frequency (kHz)
0
0.5
1
1.5
2
2.5
3
3.5
Time (frames)
0.5
1
1.5
2
2.5
3
3.5
4
4.5


Frequency (kHz)
Time (frames)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5



Frequency (kHz)
0
0.5
1
1.5
2
2.5
3
3.5
Time (frames)
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5


Frequency (kHz)
Time (frames)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5


Frequency (kHz)
0
0.5
1
1.5
2
2.5
3
3.5
Time (frames)
0.5
1
1.5
2
2.5
3
3.5
4
4.5